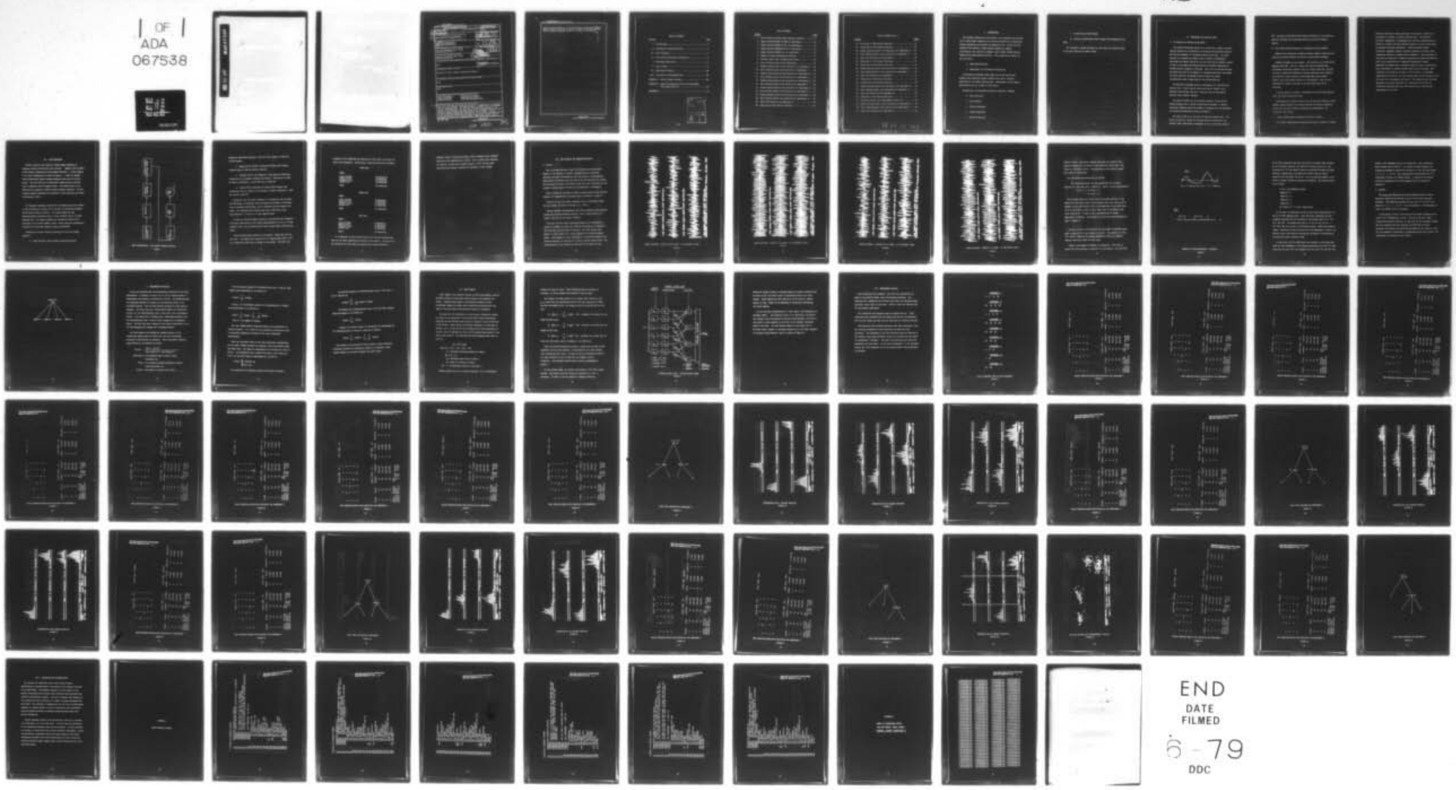
AD-A067 538 ROME AIR DEVELOPMENT CENTER GRIFFISS AFB N Y F/G 9/5
MODEM IDENTIFICATION EXPERIMENTS.(U)
FEB 79 J A FERNANDEZ
UNCLASSIFIED RADC-TR-79-55 NL

1 OF 1
ADA
067538



END
DATE
FILMED
6-79
DDC

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 14 RADC-TR-79-55	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER: 9	
4. TITLE (and Subtitle): 6 MODEM IDENTIFICATION EXPERIMENTS .	5. TYPE OF REPORT & PERIOD COVERED: [redacted] Report . for Jan [redacted] - Aug 78 ,
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): [redacted] 10 Jorge A. Fernandez	8. CONTRACT OR GRANT NUMBER(s): N/A 12 79p.
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Rome Air Development Center (ISCP) Griffiss AFB NY 13441	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 16 6[redacted] 4519[redacted] 17 ØØ
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (ISCP) Griffiss AFB NY 13441	12. REPORT DATE: 11 Feb[redacted]79
	13. NUMBER OF PAGES: 75
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Pattern Recognition	Interactive Processing
Feature Extraction	Modem Signature Analysis
Classifier Design	In-Service Performance Tests

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the design and evaluation of classifiers for distinguishing four types of modems: the CODEX LXI-9600, the HUGHES HC-276, the PARADYNE LSI-96, and LENKURT 26-C. The data used to develop these classifiers consisted of many digitized time sample waveforms for each modem and was collected by RADC's Digital Communication Experimental Facility (DICEF). With the Waveform Processing System (WPS) capabilities, the Interactive Processing Section of the Information Sciences Division (ISCP) analyzed this waveform data and extracted an initial set of eighty features. This initial set was later

DD FORM 1 JAN 73 1473

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

309 050

modified to fifty features. The On-Line Pattern Analysis and Recognition System (OLPARS) was then used to develop a number of classifier designs which are based on different subsets of the initial fifty features.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (cont...)

I. INTRODUCTION

The ultimate objective of this project is to determine the potential use of modem signal signatures as in-service indicators of transmission channel degradation and possibly as diagnostic aids. Toward this end, digitized time samples of modem signals impaired by channel perturbations were recorded on magnetic tape in RADC's DICEF (Digital Communications Experimental Facility). This objective was broken up into two parts:

a. Modem Identification.

b. Measurement of the Channel Perturbations.

The Waveform Processing System (WPS) and the On-Line Pattern Analysis and Recognition System (OLPARS) were used to identify algorithms for the modem identification. Measurement of the channel perturbations are not covered in this report.

Specifically, the following sequence of tasks was employed:

a. Data Collection

b. Data Analysis

c. Feature Hypothesis

d. Feature Extraction

e. Feature Evaluation

f. **Classification Logic Design**

g. **Testing classification logic designs with Independent Test Data.**

The classifier designs discussed in this report are based entirely on the data collected by RADC's DICEF.

II. BACKGROUND ON FACILITIES USED

A. The Waveform Processing System (WPS).

The Waveform Processing System is an interactive, graphics-oriented computer system for the extraction of features from digitized waveform data and the analysis of a digitized waveform data base. Its chief purpose is to provide the analyst with a library of mathematical algorithms and display options he can call upon from the display console so that he can design and evaluate feature extraction techniques for waveform pattern recognition problems. Once a set of features has been extracted from each of the members of a waveform data base, the analyst can input them into the OLPARS System to begin the pattern classification logic design phase of the problem solution.

The Waveform Processing System is implemented on a DEC PDP-11/45 Computer with a Vector General dislay and control console, and a Tektronix 4002 storage tube with a hardcopy unit for hardcopying selected Vector General displays.

The system includes its own executive software, filing system, display package, and a library of application programs. A feature extraction language allows the analyst to construct his own algorithms for waveform processing and feature extraction.

The input to WPS is in the form of digitized waveform data. The system is built as a series of overlays which are callable by the operator from a menu which is displayed to him on the Vector General

CRT. The data in the form of data trees is available to the analyst by means of utilizing the interactive devices on the Vector General console.

B. The On-Line Pattern Analysis and Recognition System (OLPARS).

OLPARS is an interactive, graphics-oriented computer system for the solution of pattern analysis and pattern classification problems.

OLPARS is resident on two systems. One version is on the PDP-11/45 Computer under WPS. This is a single user system employing high performance interactive graphics, and, as a module under WPS, provides for ease of interaction between the feature hypothesis mode conducted under WPS and a rapid testing of these hypothesis under OLPARS. However, since this system resides on a minicomputer, there are core limitations in terms of the size of the data base which can be processed.

A second version of OLPARS is implemented on the HIS 6180 Computer under the MULTICS Operating System.

Both versions of OLPARS include their own executive software, filing system, display package, and software modules for feature evaluation, vector data structure analysis, measurement transformation, and classifier logic design.

C. Digital Communications Experimental Facility (DICEF).

The Digital Communications Experimental Facility (DICEF) is a unique

laboratory dedicated to data acquisition and analysis, research and development in support of digital communications. This facility provides a combination of programmed data reduction capabilities and a variety of in-place real and simulated channels to allow a wide choice of equipment and media experiments. Media simulators provide controlled, repeatable channel conditions essential to conduct valid comparative analysis of communications equipments. Units evaluated in this mode are subjected to numerous combinations of known perturbations which can be controlled in a completely deterministic manner. Regardless of whether real or synthetic media are used, correlation of error performance to channel characteristics can be obtained. This capability is provided by the heart of the facility - a high-speed communications processor, the 9303 Message Switch, which operates at any data rate up to ten megabits per second. The communications processor possesses the critical attribute of a high-speed I/O so that all information regarding high data rate channels can be acquired and manipulated in real time.

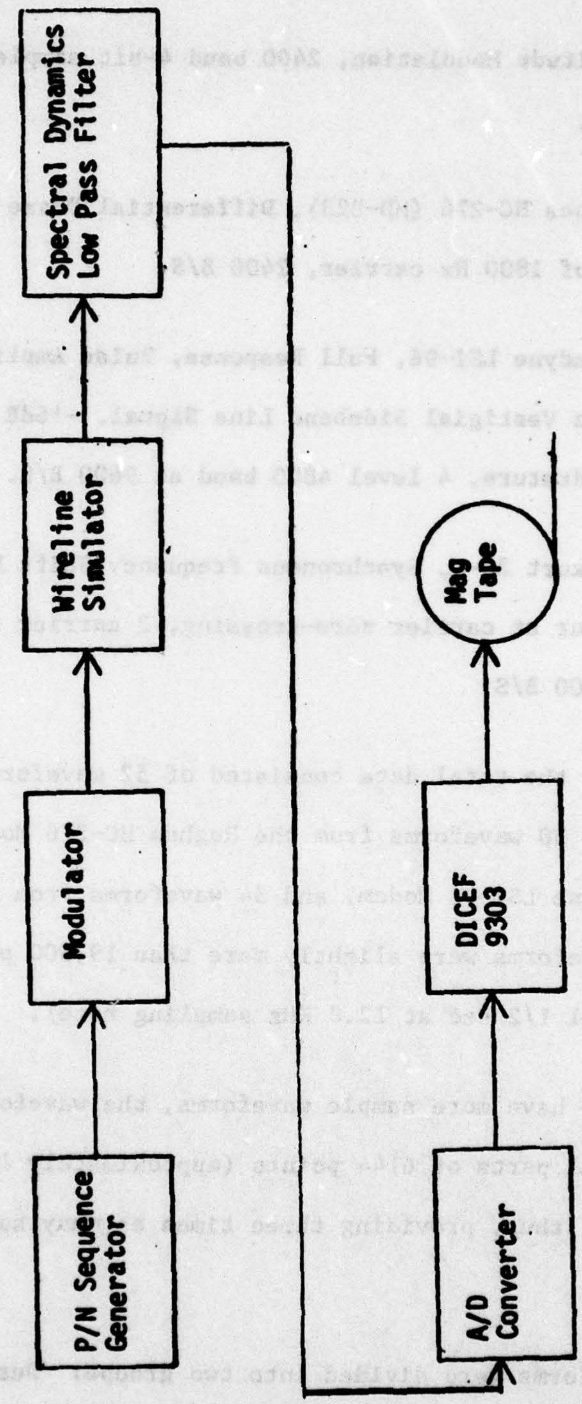
III. DATA COLLECTION

Several digitized time samples of modem signals impaired by telephone channel perturbations were recorded on magnetic tape in RADC's DICEF (Digital Communications Experimental Facility). A block diagram of the test configuration is shown in Figure 1. A 2047 bit maximal length psuedorandom digital sequence generator was used as the data source. The unit used was II Corporation's BERT 901 with an RS-232 output compatible with all modems tested. The analog output of the modem was then applied to DICEF's Wireline Channel Simulator. The two technical manuals describing the operation of this simulator are listed as References 4 and 5.

All frequency components beyond half the sampling rate were removed by the anti-aliasing low pass filter portion of the Spectral Dynamics SD-360 Digital Signal Processor. The analog signal was then Analog-to-Digital converted using a 12-bit converter with a 12.8 KHz sampling rate. The digital samples were recorded by DICEF's 9303 processor on its 7-track magnetic tapes. These tapes were subsequently converted to 9-track WPS compatible tapes by RADC/ISCP.

Following is a short technical description of the four modems employed:

a. Codex LSI-9600, Double Sideband Suppressed Carrier



TEST CONFIGURATION - AFCS MODEM SIGNATURE ANALYSIS

FIGURE 1

Quadrature Amplitude Modulation, 2400 baud 4-bit samples at 9600 B/S, 1706 Hz Carrier.

b. Hughes HC-276 (MD-823), Differential Phase Shift Keying, 4-phase keying of 1800 Hz carrier, 2400 B/S.

c. Paradyne LSI-96, Full Response, Pulse Amplitude Modulation transmitted as a Vestigial Sideband Line Signal, -16dB carrier at 2853 Hz added in quadrature, 4 level 4800 baud at 9600 B/S.

d. Lenkurt 26-C, Synchronous Frequency Shift Keying, data transitions occur at carrier zero-crossing, 2 carrier frequencies - 1200 and 2400 Hz, 1200 B/S.

Originally, the total data consisted of 52 waveforms from the Codex LSI-9600 Modem, 30 waveforms from the Hughes HC-276 Modem, 35 waveforms from the Paradyne LSI-96 Modem, and 34 waveforms from the Lenkurt 26-C Modem. The waveforms were slightly more than 19,800 points long (approximately 1 1/2 sec at 12.8 KHz sampling rate).

In order to have more sample waveforms, the waveforms were segmented into three equal parts of 6144 points (approximately 1/2 sec at 12.8 KHz sampling rate), thus, providing three times as many sample waveforms per modem.

These waveforms were divided into two groups: Design Data and the Test Data. The Design Data consisted of 227 (approximately 50%) of the 453 waveforms and were used to design the classifier. The Test Data

consisted of the remaining 226 waveforms and were later to be used as a test of the classifier. Specifically, these data sets are as follows:

TOTAL DATA

MODEM	
Codex LSI-9600	156 Waveforms
Hughes HC-276	90 Waveforms
Paradyne LSI-96	105 Waveforms
Lenkurt 26-C	102 Waveforms
TOTAL	453 Waveforms

DESIGN DATA

MODEM	
Codex LSI-9600	78 Waveforms
Hughes HC-276	45 Waveforms
Paradyne LSI-96	53 Waveforms
Lenkurt 26-C	51 Waveforms
TOTAL	227 Waveforms

TEST DATA

MODEM	
Codex LSI-9600	78 Waveforms
Hughes HC-276	45 Waveforms
Paradyne LSI-96	52 Waveforms
Lenkurt 26-C	51 Waveforms
TOTAL	226 Waveforms

It is important to have in mind that all of the data collected was used for the modem identification portion of the effort. All but one of the waveforms for each modem was impaired with varying degrees of

different types of distortions (phase jitter, Gaussian noise, harmonic distortion, and combinations of these). As our results will indicate, the features utilized were powerful enough to "see" through these distortions and properly classify the waveforms' origin (modem).

IV. DATA ANALYSIS AND FEATURE EXTRACTION

A. General.

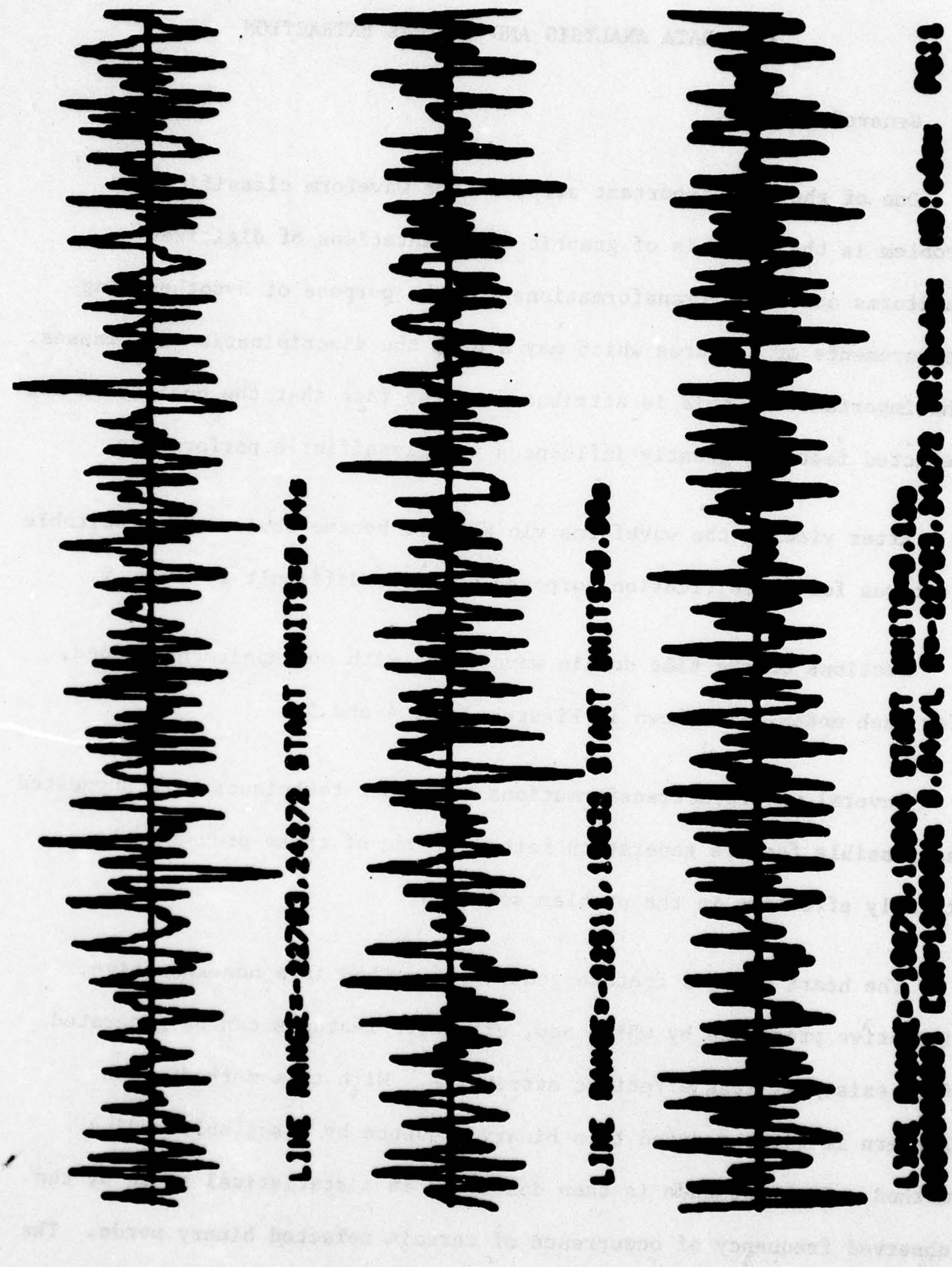
One of the most important steps in the waveform classification problem is the analysis of graphic representations of digitized waveforms and their transformations for the purpose of hypothesizing measurements or features which may aid in the discrimination of classes. The importance of this is attributed to the fact that the quality of the selected features greatly influences the classifier's performance.

After viewing the waveforms via WPS, it became obvious that suitable features for classification purposes would be difficult to come by.

Sections of the time domain waveforms, with no impairments added, for each modem, are shown in Figures 2, 3, 4 and 5.

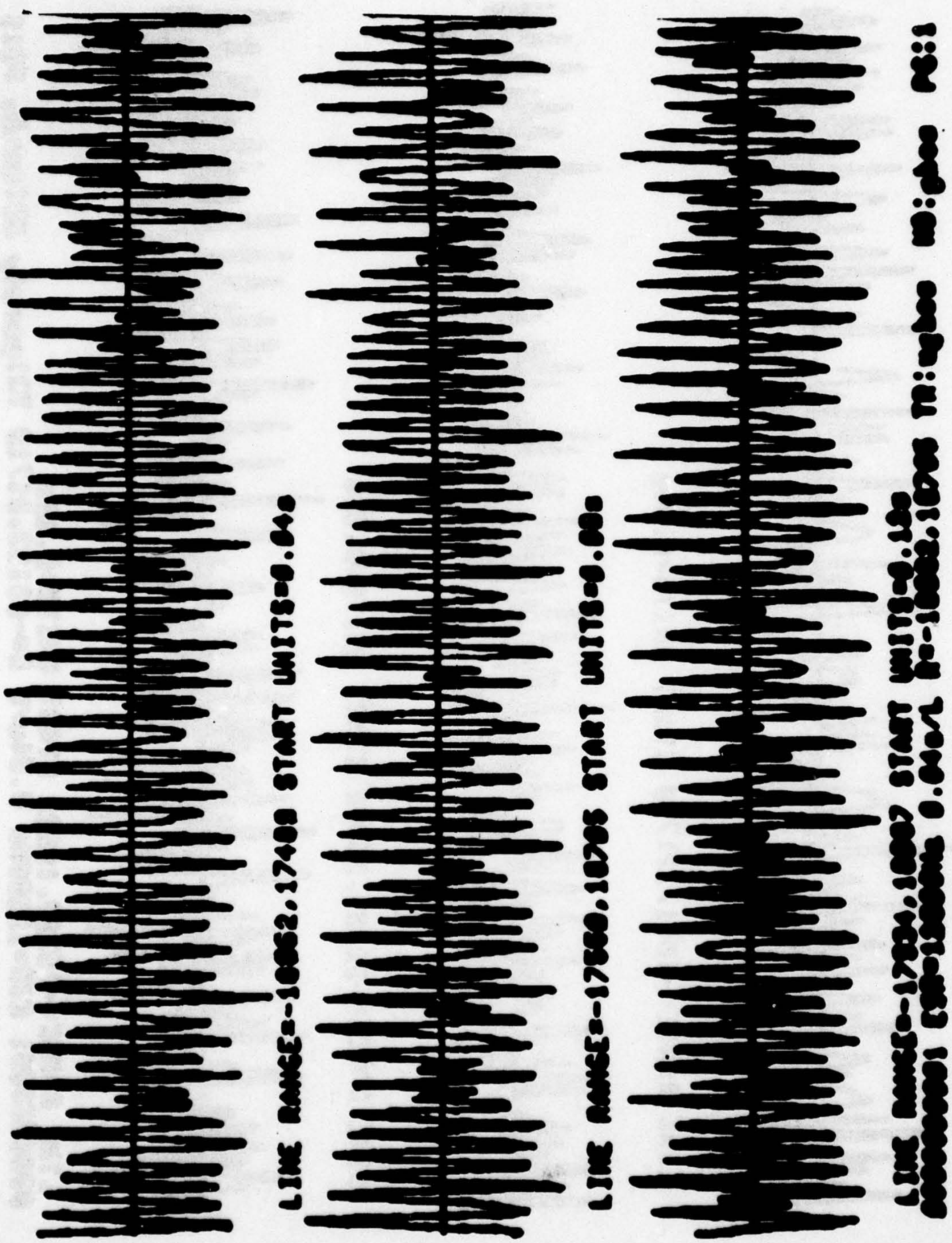
Several waveform transformations and other techniques were suggested as possible feature generation methods. One of these proved to be totally effective in the problem solution.

The heart of this feature generation method is a nonexhaustive, iterative procedure by which new, effective features can be generated from existing, less effective attributes. With this method, each pattern is first reduced to a binary sequence by a suitable coding method. Each sequence is then described in a statistical sense by the observed frequency of occurrence of certain selected binary words. The categorization of the patterns is performed on the basis of such



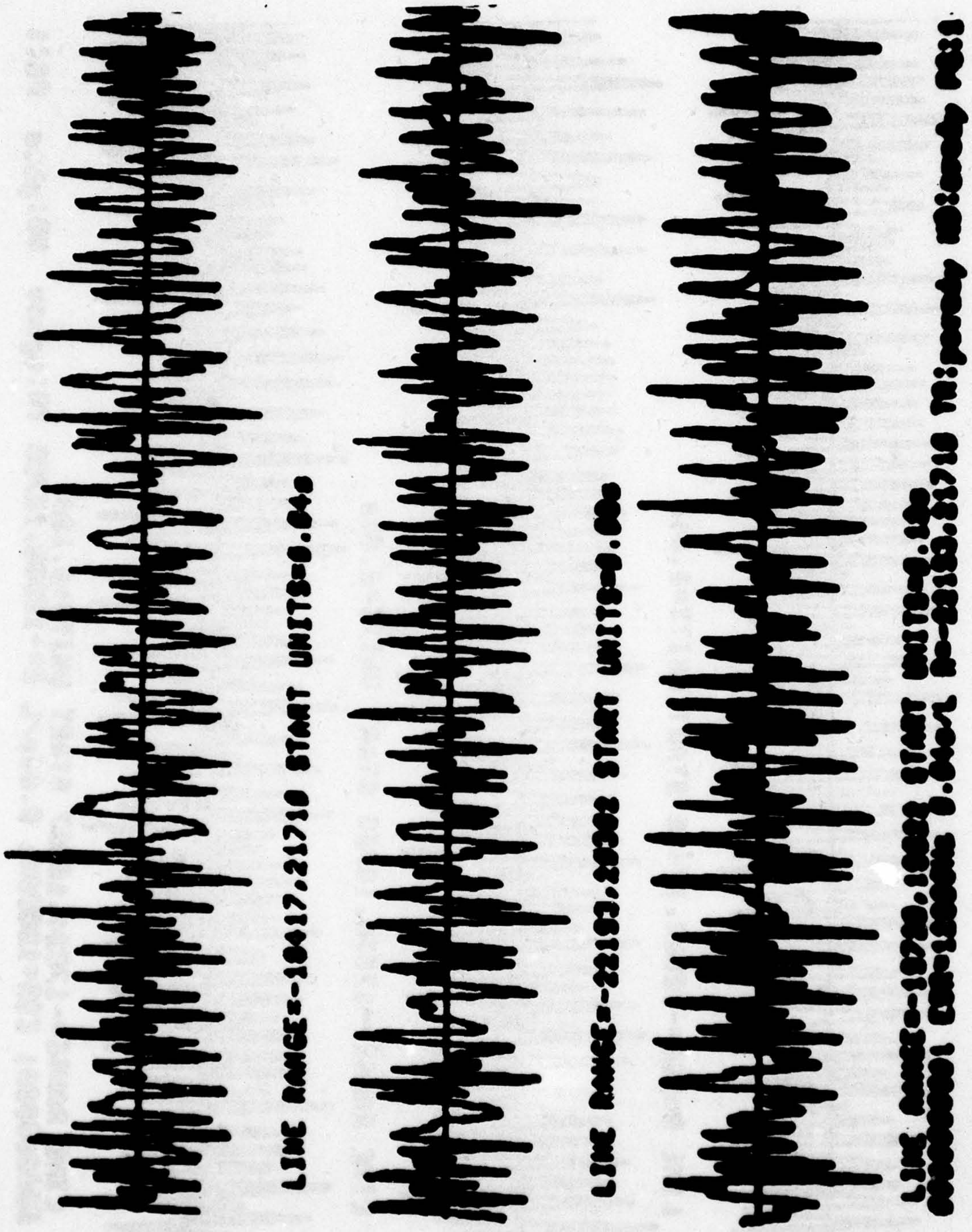
SAMPLE WAVEFORM - CODEX LSI-9600 MODEM - NO IMPAIRMENTS ADDED

FIGURE 2



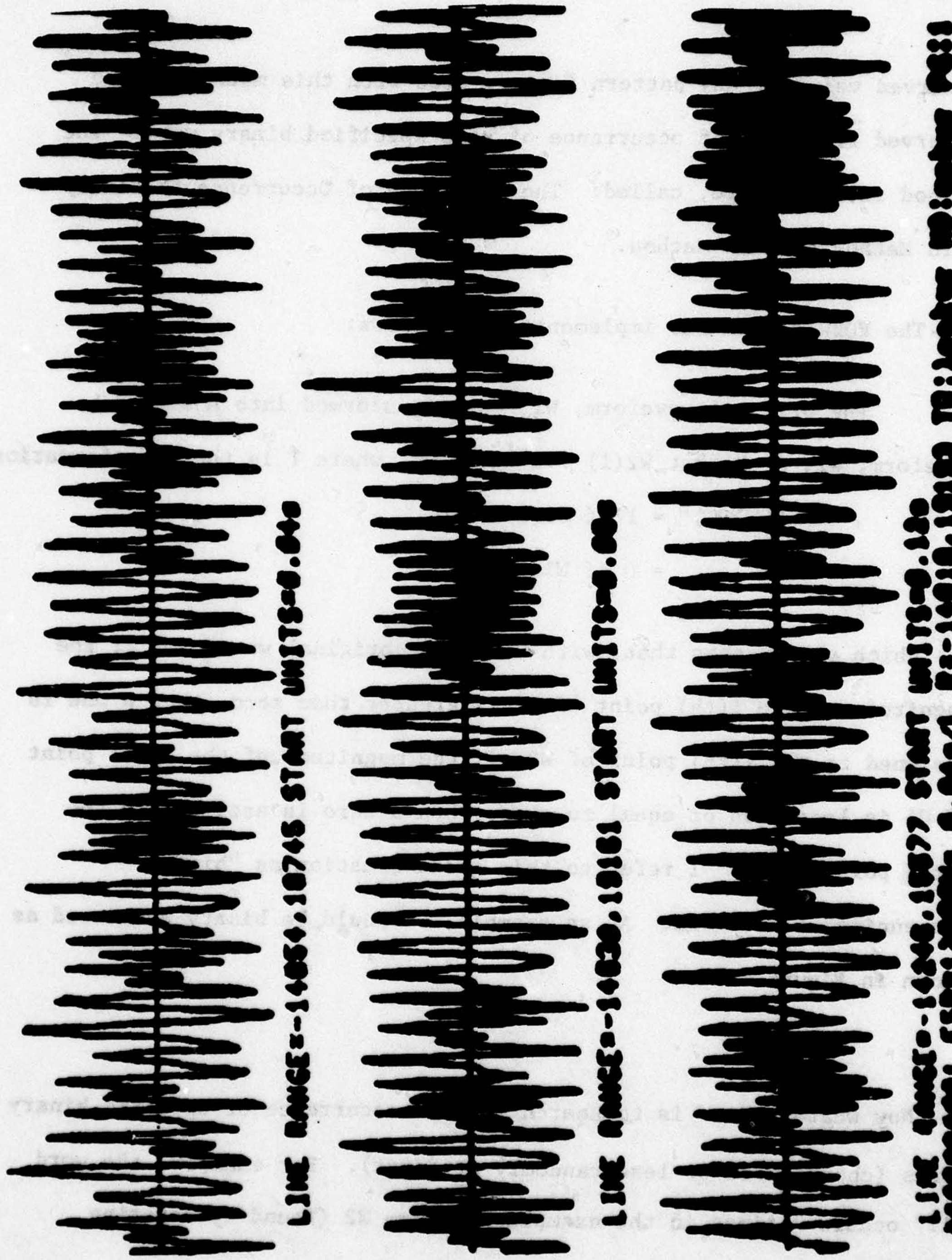
SAMPLE WAVEFORM - HUGHES HC-276 MODEM - NO IMPAIRMENTS ADDED

FIGURE 3



SAMPLE WAVEFORM - PARADYNE LSI-96 MODEM - NO IMPAIRMENTS ADDED

FIGURE 4



SAMPLE WAVEFORM - LENKURT 26-C MODEM - NO IMPAIRMENTS ADDED

FIGURE 5

observed values. Any pattern feature used with this method is the observed frequency of occurrence of some specified binary word. The method is, therefore, called: The Frequency of Occurrence of Binary Word Method or FOBW Method.

The FOBW Method was implemented as follows:

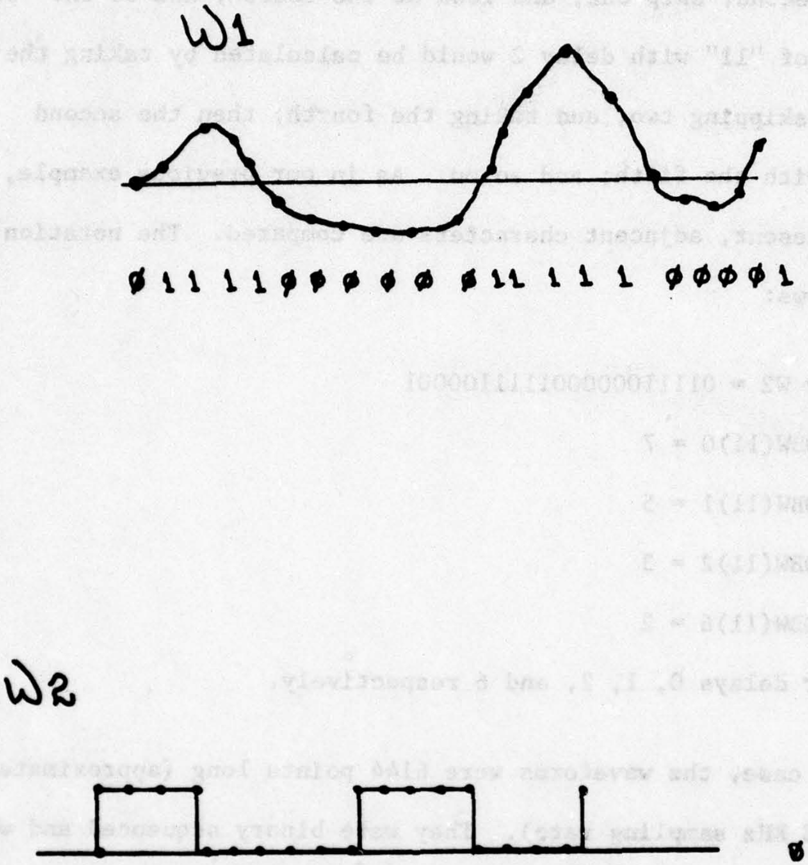
The original waveform, W1, was transformed into a second waveform, W2, such that $W2(i) = T[W1(i)]$, where T is the transformation:

$$T[W1(i)] = 1 \text{ if } W1(i) > 0$$
$$= 0 \text{ if } W1(i) \leq 0$$

Which simply says that, with W1 as the original waveform, if the magnitude of the i(th) point of W1 is greater than zero, then a one is assigned to the i(th) point of W2; if the magnitude of the i(th) point of W1 is less than or equal to zero, then a zero is assigned to the i(th) point of W2. I refer to this transformation as "binary sequencing" a waveform. As an example, W1 would be binary sequenced as shown in Figure 6.

Now what is done is to search for the occurrence of specific binary words (chosen more or less randomly at first). For example, the word "11" occurs 7 times in the example waveform W2 (found by counting adjacent characters which are both ones).

Finally, the concept of "delays" is introduced. If we were to search for the occurrence of the word "11" with delay 1, we would look



EXAMPLE OF BINARY-SEQUENCING A WAVEFORM

FIGURE 6

at the first character, skip one, and look at the third; then we would go to the second, skip one, and look at the fourth; and so on. The occurrence of "11" with delay 2 would be calculated by taking the first character, skipping two, and taking the fourth; then the second character with the fifth; and so on. As in our previous example, if no delay is present, adjacent characters are compared. The notation would be as follows:

for W2 = 01111000000111110000l

FOBW(11)0 = 7

FOBW(11)1 = 5

FOBW(11)2 = 3

FOBW(11)6 = 2

for delays 0, 1, 2, and 6 respectively.

In our case, the waveforms were 6144 points long (approximately 1/2 sec at 12.8 KHz sampling rate). They were binary sequenced and were originally searched for the occurrence of the binary words "11", "10", "01", and "00". The delays used were 0, 10, 20, 30, 40, 50, 60, ..., 170, 180, 190; for a total of 20 different delays. These were chosen at random. Resulting for each waveform was an 80 dimensional (4 words x 20 delays) vector; each feature representing the number of occurrences of the given word with a specific delay.

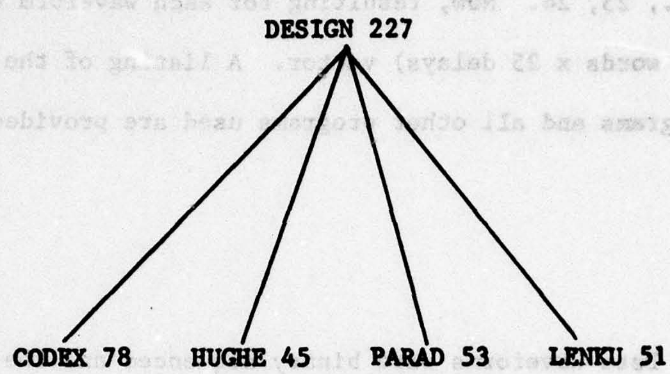
As experience with the FOBW Method was acquired, it was noted that there was some redundancy in the features obtained by the word "11" with those from the word "00"; and likewise with the words "10" and "01" (in

general, this redundancy is not to be expected). Also, potentially better delays were arrived at. As a result of these findings, the program was modified to search for the words "11" and "10" with delays 0, 1, 2, 3, ..., 23, 24. Now, resulting for each waveform was a 50 dimensional (2 words x 25 delays) vector. A listing of the feature extraction programs and all other programs used are provided in Appendix A.

B. Specific.

The Design Data Waveforms were binary sequenced and the feature extraction algorithm was executed on each of the binary sequenced waveforms. This algorithm searches for the words "11" and "10" with delays 0, 1, 2, 3, ..., 23, 24. For each waveform, a 50 dimensional (2 words x 25 delays) vector is computed.

At this stage, we have a tree that has four nodes (classes) with a total of 227 50 dimensional vectors. This set of vectors, which contains the extracted features, is then used for the classifier design. Once the classifier has been achieved, the TEST DATA is binary sequenced, the features are extracted and these are then used as a test for the classifier's efficiency in discriminating the four classes. The design data tree structure is as follows:


DESIGN 227
CODEX 78
HUGHE 45
PARAD 53
LENKU 51

V. MEASUREMENT EVALUATION

We are now concerned with the discriminatory qualities of our fifty measurements. In general, we would like to use the minimum number of measurements that achieves a satisfactory solution. The OLPARS provides two suboptimal methods for ranking the discriminatory power of the extracted features. Each of these methods provides for three types of rankings. The first type uses a significance measure of a particular feature, xp, for discriminating class i from class j and is designated Mij(xp). The second type of ranking uses a significance measure of xp for discriminating class i from all other classes and is designated Mi(xp). The last type uses a measure of the overall significance of xp for discriminating all classes and is designated M(xp).

The first method on the OLPARS for ranking features is the discriminant measure which is useful when the class conditional probability distributions are unimodal. These discriminant measures, using feature xp, are defined as follows:

$$\mathrm{Mij}(xp) = \frac{[\bar{x}p(i) - \bar{x}p(j)]^2}{[(Ni-1)(\hat{\sigma}p(i))^2 + (Nj-1)(\hat{\sigma}p(j))^2]}$$

where $\bar{x}p(j)$ = the estimated mean of class j along measurement xp,

$\hat{\sigma}p(j)$ = the estimated standard deviation of class j along measurement xp,

and Ni = the number of vectors from class i.

The discriminant measure for differentiating class i from all other classes using measurement xp is defined as:

$$Mi(xp) = \sum_{j \neq i}^{K} Mij(xp)$$

Finally, the discriminant measure for distinguishing all classes using measurement xp is defined as:

$$M(xp) = \sum_{i=1}^{K} Mi(xp) = \sum_{i=1}^{K} \sum_{j \neq i}^{K} Mij(xp)$$

where K = the number of classes.

The other OLPARS feature evaluation method is the probability of confusion measure. It is valid for any probability distribution since it essentially measures the overlap of the class conditional probabilities.

Since the functional forms of the class conditional probabilities are not known, OLPARS estimates the marginal class distributions using the sample data. The range for measurement xp is divided into cells of width Δ. The probability that a sample from class j will occupy the r(th) cell along the range of measurement xp is given by:

$$Prp(j) = \int_{r(th)\ cell} P(xp/Cj)\, dxp$$

The probability of confusion measures are defined as follows:

The pairwise measure for differentiating class i from class j can be computed by:

$$Mij(xp) = \sum_{r=1}^{N_p} \min_{i,j} (Prp(i), Prp(j))$$

The measure for differentiating class i from all other classes using measurement xp is defined by:

$$Mi(xp) = \sum_{j \neq i}^{K} Mij(xp)$$

Finally, the overall measure of significance of measurement xp for differentiating all classes is computed as follows:

$$M(xp) = \sum_{i=1}^{K} Mi(xp) = \sum_{i=1}^{K} \sum_{j \neq i}^{K} Mij(xp)$$

The ranking of the extracted features based on these evaluation techniques provides the information required to rationally choose initial subsets of the fifty features for logic design.

VI. LOGIC DESIGN

Logic design is an iterative process in which many designs, based on modified versions of the initial feature subsets, are generated and tested. Features which appear to discriminate between the more troublesome classes are added, while superfluous features which rank high for the same easily discriminated classes are eliminated.

The logic for the classifiers for this pattern recognition problem are based on two approaches: the Pairwise Fisher Linear Discriminant Technique and User-Defined Logic based on coordinate vector projections. In the Pairwise Fisher Linear Discriminant Technique, for each pair of classes i and j a unit vector dij is computed such that projections of the data onto dij maximize the ratio of the between-class scatter to the within-class scatter. The direction dij which maximizes this ratio is given by:

$$d_{ij} = \alpha W^{-1}_{ij} \underset{\sim}{\Delta}_{ij}$$

where $W_{ij} = (N_i - 1)C_i + (N_j - 1)C_j$

C_i = Estimated Covariance Matrix for class i

$\underset{\sim}{\Delta}_{ij} = \underset{\sim}{\mu}_i - \underset{\sim}{\mu}_j$

$\underset{\sim}{\mu}_i$ = Estimated mean vector of class i

N_i = Number of vectors in class i

and α = Normalizing constant so that $|\underset{\sim}{d}| = 1$

OLPARS computes dij and an initial threshold, θ_{ij}, to distinguish

between all pairs of class. These thresholds may be adjusted, if necessary, to obtain optimal discrimination along each $\underset{\sim}{d}ij$.

For example, the inner product of an unknown input feature vector, $\underset{\sim}{x}$, is taken with the discriminant dCH for the pair consisting of CODEX LSI-9600 and HUGHES HC-276, and compared with the threshold θCH for that pair.

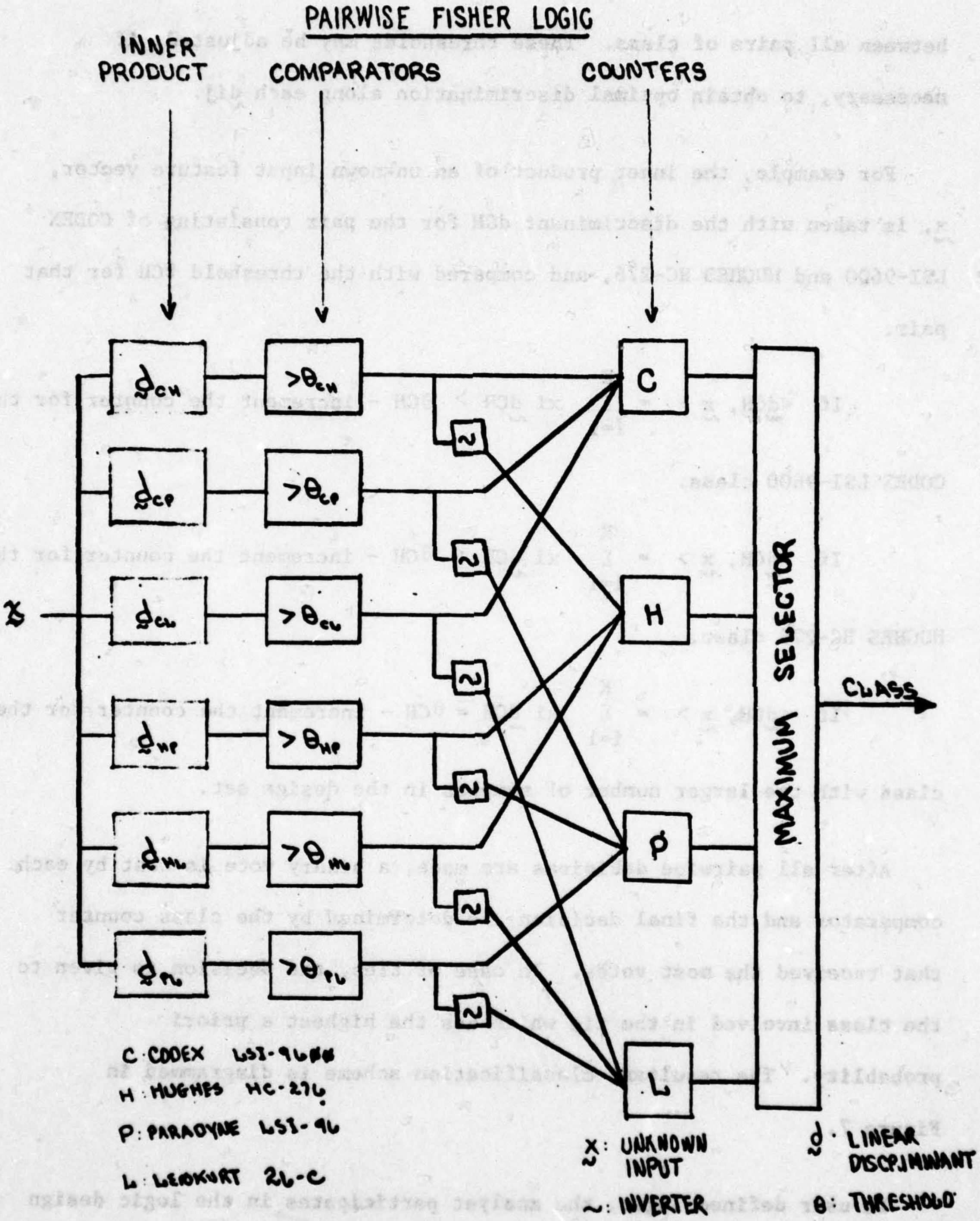
If $\langle \underset{\sim}{d}CH, \underset{\sim}{x} \rangle = \sum_{i=1}^{K} xi\ \underset{\sim}{d}CH > \theta CH$ - increment the counter for the CODEX LSI-9600 class.

If $\langle \underset{\sim}{d}CH, \underset{\sim}{x} \rangle = \sum_{i=1}^{K} xi\ \underset{\sim}{d}CH < \theta CH$ - increment the counter for the HUGHES HC-276 class.

If $\langle \underset{\sim}{d}CH, \underset{\sim}{x} \rangle = \sum_{i=1}^{K} xi\ \underset{\sim}{d}CH = \theta CH$ - increment the counter for the class with the larger number of samples in the design set.

After all pairwise decisions are made, a binary vote is cast by each comparator and the final decision is determined by the class counter that received the most votes. In case of ties, the decision is given to the class involved in the tie which has the highest a priori probablity. The resultant classification scheme is diagrammed in Figure 7.

In user defined logic, the analyst participates in the logic design process. The vectors from the classes are projected on a one- or two-space. If there is (in the analyst's judgment) sufficient



PAIRWISE FISHER LOGIC - CLASSIFICATION SCHEME

FIGURE 7

separation between classes, or between groups of classes, boundaries may be drawn so that the feature space is partitioned into two or three regions. These regions are then labeled as to the class or classes present in them. Figure 36 in Experiment 8 illustrates partitioning into three regions.

For the one-space implementation of these logics, the mathematics is extremely simple. The unlabeled vector to be classified is projected (dot product) onto the projection direction (discriminant); the value of this scalar is then compared to the value of the boundary (threshold drawn by the user). All user defined logics in this report are in one-space logic, however, a two-space scatter plot of the logic designed in one-space using features 4 and 27 is given in Figure 37.

VII. EXPERIMENTAL RESULTS

Nine classifiers were designed. The first four classifiers are based on the pairwise Fisher Linear Discriminant Technique. The remaining five classifiers are decision trees which use one-dimensional coordinate vector logic at each node. Figure 8 lists the features used for each design (experiment).

All classifiers were designed using the Design Data set. These classifiers were evaluated with the Design Data set and an independent Test Data set (these two data sets are described in a previous section).

The Design and Test confusion matrices, with their statistics, from the resulting evaluation of each classifier are given for each experiment. The histograms of the data projected along the features in the decision trees using coordinate vectors in one-space are also given for Experiments 5 through 9. The logic tree structures for these five experiments are also shown. In the case of Experiment 8, the two-space scatter plot with reference to the two features used in that experiment is included.

EXPERIMENT 1

5 9 15 25

EXPERIMENT 2

30 34 40 50

EXPERIMENT 3

2 5 9 21

EXPERIMENT 4

27 30 34 46

EXPERIMENT 5

29 30 34

EXPERIMENT 6

27 34 46

EXPERIMENT 7

3 8 27

EXPERIMENT 8

4 27

EXPERIMENT 9

27 29

LIST OF FEATURES USED IN EACH EXPERIMENT

FIGURE 8

LOGIC TREE zzz   DATA TREE zzz

	U	D	E	X	*
U	51	0	0	0	0
D	0	53	0	0	0
E	0	0	45	0	0
X	0	0	0	78	0

LOGIC TREE zzz   DATA TREE zzz

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
U	51	51	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
X	78	78	100.00%	0	0.00%	0	0.00%

```
TOTAL    VECTORS      227
OVERALL  CORRECT      227 FOR 100.00/
OVERALL  ERROR          0 FOR   0.00/
OVERALL  REJECTED       0 FOR   0.00/
```

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 1

FIGURE 9

LOGIC TREE qqq DATA TREE qqq

	U	D	E	X	*
U	51	0	0	0	0
D	0	51	0	1	0
E	0	0	45	0	0
X	0	1	0	77	0

LOGIC TREE qqq DATA TREE qqq

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
U	51	51	100.00%	0	0.00%	0	0.00%
D	52	51	98.07%	1	1.92%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
X	78	77	98.71%	1	1.28%	0	0.00%

TOTAL VECTORS	226		
OVERALL CORRECT	224	FOR	99.11/
OVERALL ERROR	2	FOR	0.88/
OVERALL REJECTED	0	FOR	0.00/

TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 1

FIGURE 10

LOGIC TREE xxx DATA TREE xxx

	U	D	E	X	*
U	51	0	0	0	0
D	0	53	0	0	0
E	0	0	45	0	0
X	0	0	0	78	0

LOGIC TREE xxx DATA TREE xxx

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
U	51	51	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
X	78	78	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 227
OVERALL CORRECT 227 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 2

FIGURE 11

LOGIC TREE sss DATA TREE sss

	U	D	E	X	*
U	51	0	0	0	0
D	0	52	0	0	0
E	0	0	45	0	0
X	0	0	0	78	0

		LOGIC TREE sss		DATA TREE sss			
CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
U	51	51	100.00%	0	0.00%	0	0.00%
D	52	52	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
X	78	78	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 226
OVERALL CORRECT 226 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 2

FIGURE 12

LOGIC TREE rrr DATA TREE rrr

	U	D	E	X	*
U	51	0	0	0	0
D	0	53	0	0	0
E	0	0	45	0	0
X	0	0	0	78	0

LOGIC TREE rrr DATA TREE rrr

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
U	51	51	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
X	78	78	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 227
OVERALL CORRECT 227 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 3

FIGURE 13

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

	LOGIC TREE sss					DATA TREE sss
	U	D	E	X	*	
U	51	0	0	0	0	
D	0	52	0	0	0	
E	0	0	45	0	0	
X	0	0	0	78	0	

		LOGIC TREE sss		DATA TREE sss			
CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
U	51	51	100.00%	0	0.00%	0	0.00%
D	52	52	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
X	78	78	100.00%	0	0.00%	0	0.00%

```
TOTAL    VECTORS     226
OVERALL  CORRECT     226 FOR 100.00/
OVERALL  ERROR         0 FOR   0.00/
OVERALL  REJECTED      0 FOR   0.00/
```

TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 3

FIGURE 14

LOGIC TREE www DATA TREE www

	U	D	E	X	*
U	51	0	0	0	0
D	0	53	0	0	0
E	0	0	45	0	0
X	0	0	0	78	0

LOGIC TREE www DATA TREE www

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
U	51	51	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
X	78	78	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS	227	
OVERALL CORRECT	227 FOR	100.00/
OVERALL ERROR	0 FOR	0.00/
OVERALL REJECTED	0 FOR	0.00/

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 4

FIGURE 15

LOGIC TREE vvv DATA TREE vvv

	U	D	E	X	*
U	51	0	0	0	0
D	0	52	0	0	0
E	0	0	45	0	0
X	0	0	0	78	0

LOGIC TREE vvv DATA TREE vvv

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
U	51	51	100.00%	0	0.00%	0	0.00%
D	52	52	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
X	78	78	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 226
OVERALL CORRECT 226 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 4

FIGURE 16

LOGIC TREE HHH DATA TREE DESIGN

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	53	0	0
U	0	0	0	51	0

LOGIC TREE HHH DATA TREE DESIGN

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 227
OVERALL CORRECT 227 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 5

FIGURE 17

LOGIC TREE HHH DATA TREE TEST

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	1	0	51	0	0
U	0	0	0	51	0

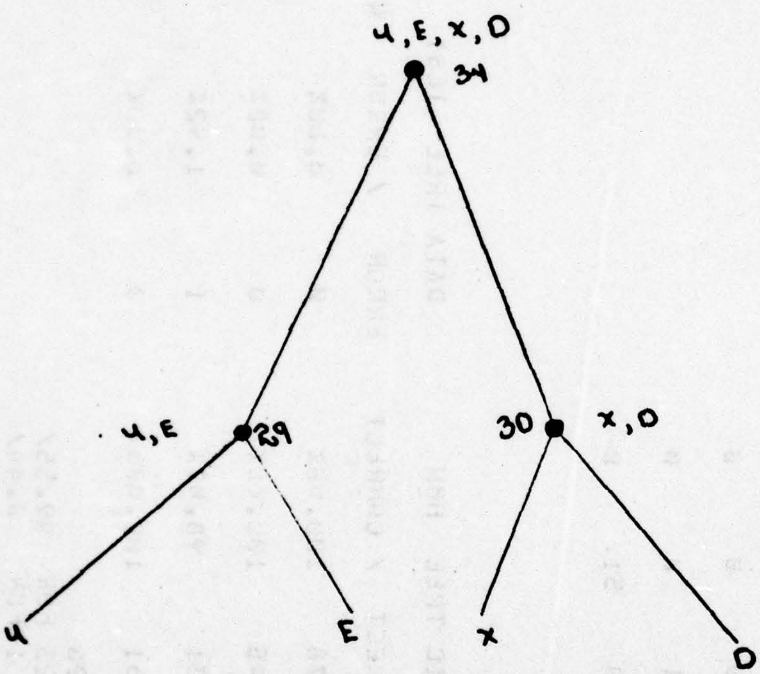
LOGIC TREE HHH DATA TREE TEST

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	52	51	98.07%	1	1.92%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 226
OVERALL CORRECT 225 FOR 99.55/
OVERALL ERROR 1 FOR 0.44/
OVERALL REJECTED 0 FOR 0.00/

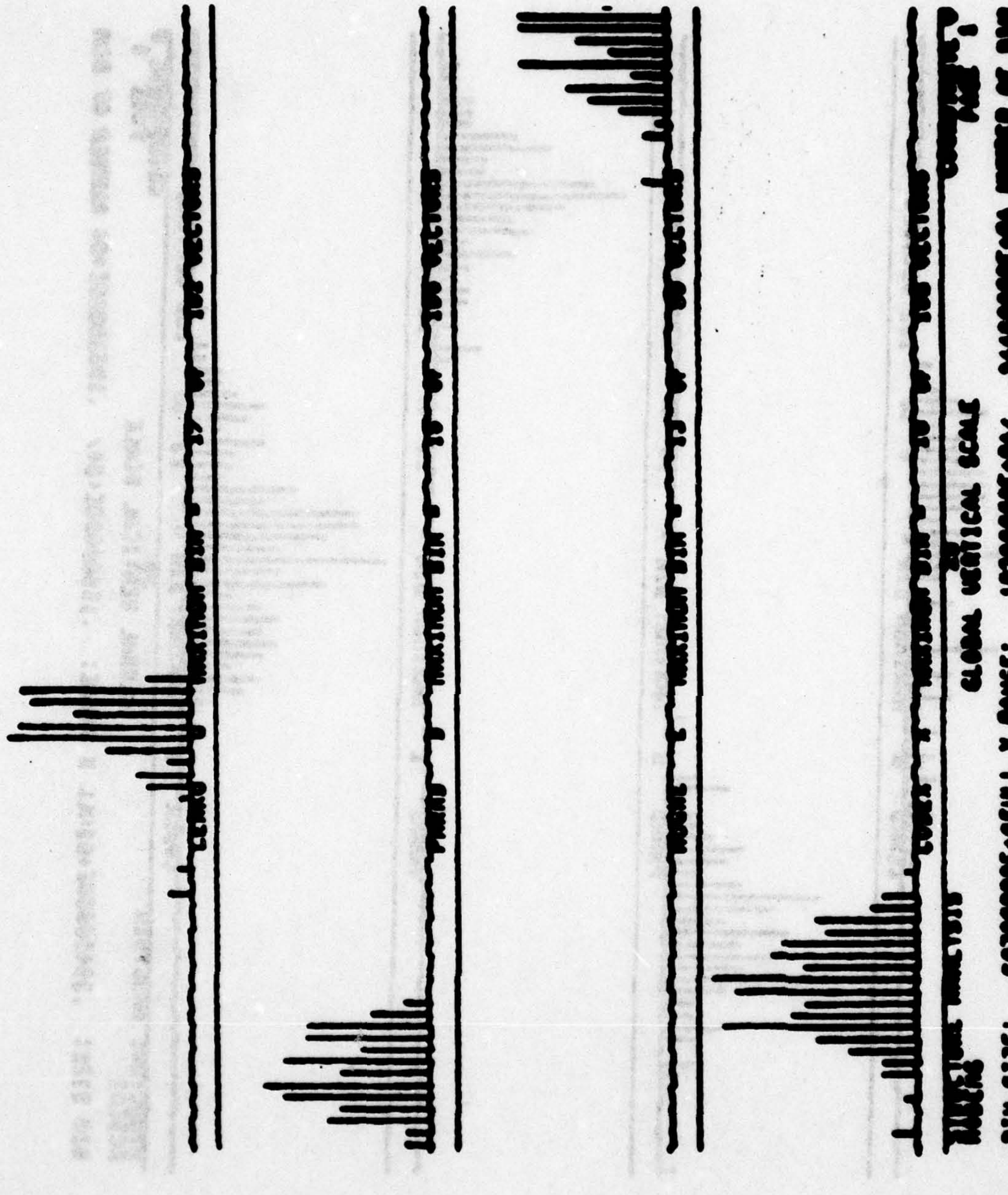
TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 5

FIGURE 18



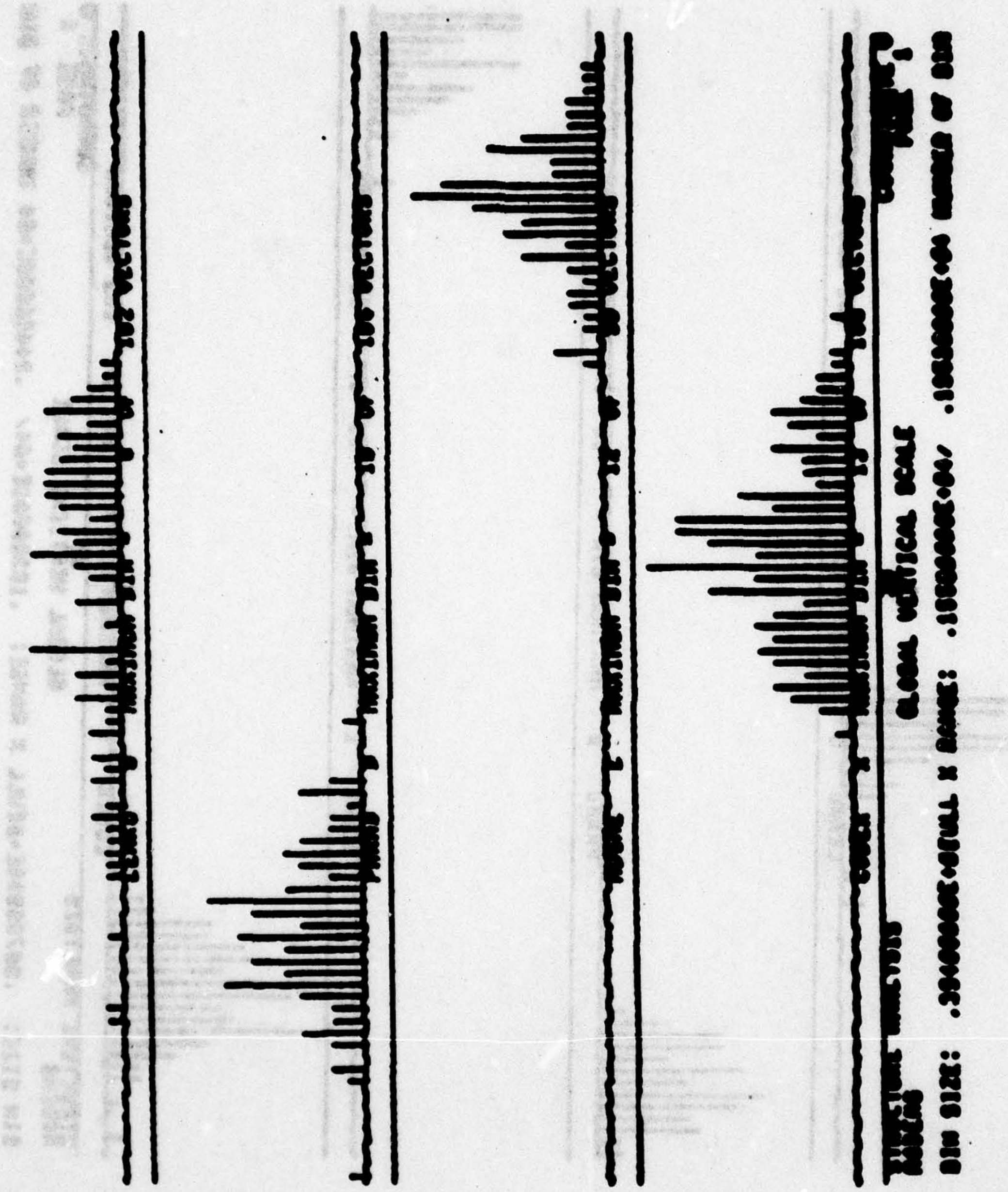
LOGIC TREE STRUCTURE FOR EXPERIMENT 5

FIGURE 19



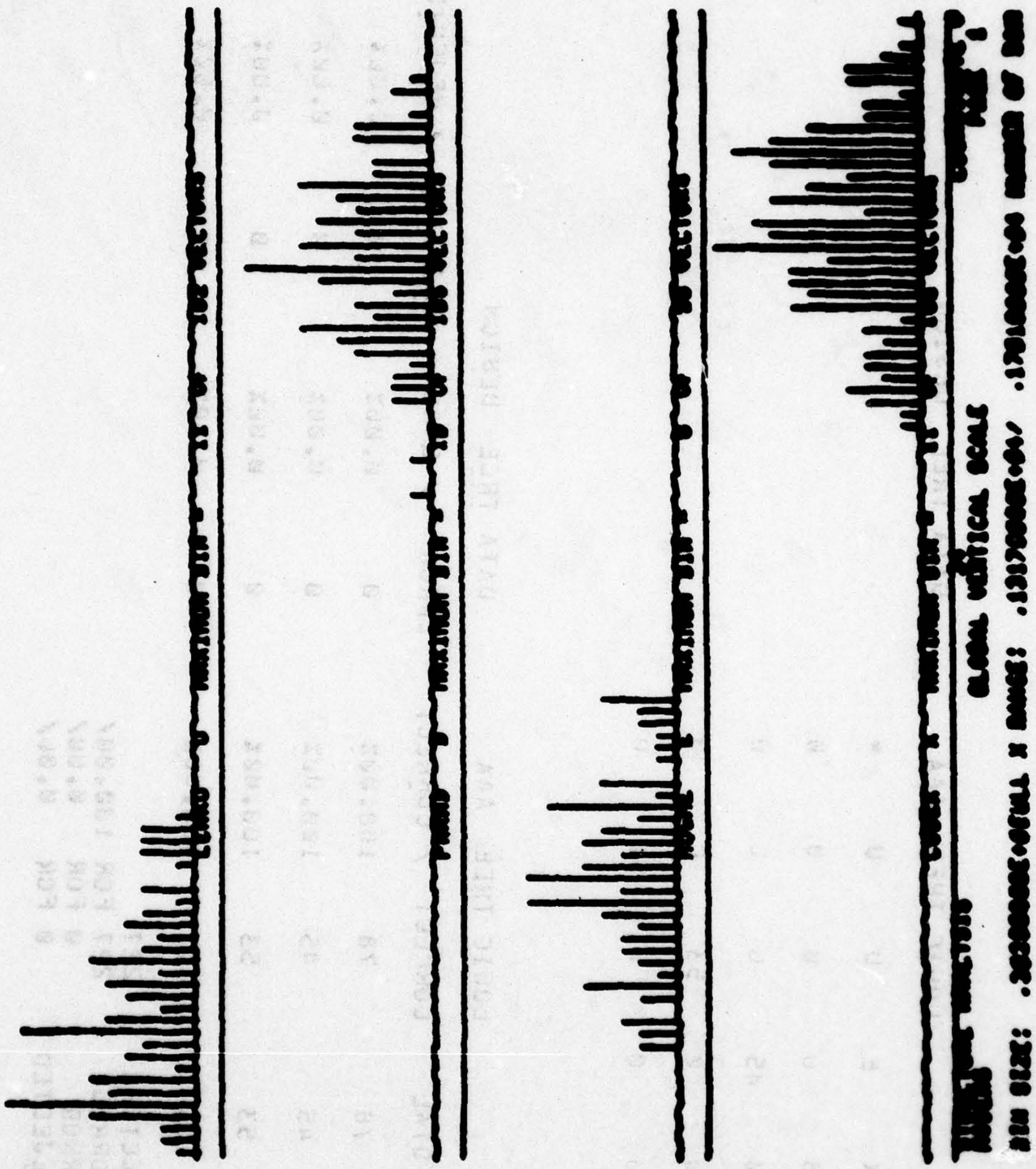
PROJECTION ON 29th FEATURE DIRECTION

FIGURE 20



PROJECTION ON 30th FEATURE DIRECTION

FIGURE 21



PROJECTION ON 34th FEATURE DIRECTION

FIGURE 22

LOGIC TREE AAA DATA TREE DESIGN

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	53	0	0
U	0	0	0	51	0

LOGIC TREE AAA DATA TREE DESIGN

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 227
OVERALL CORRECT 227 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 6

FIGURE 23

LOGIC TREE AAA   DATA TREE TEST

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	52	0	0
U	0	0	0	51	0

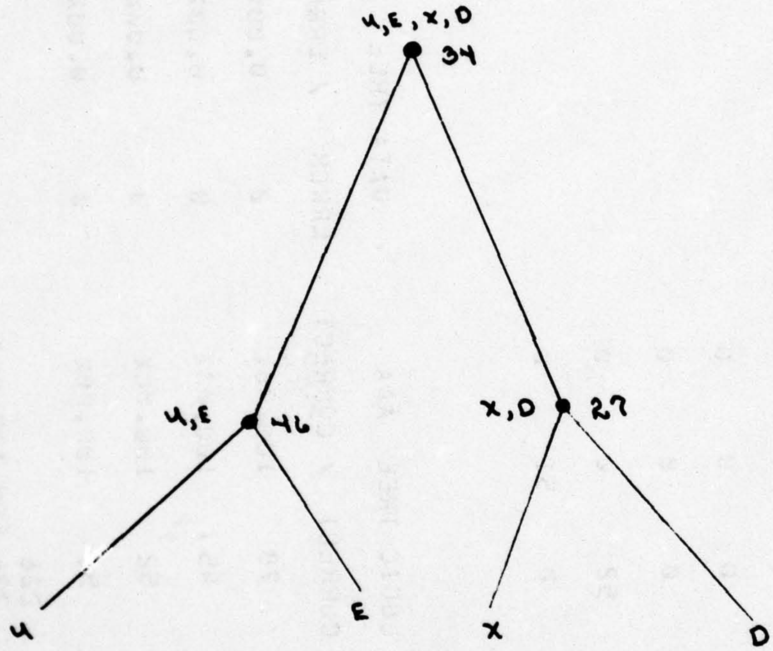
LOGIC TREE AAA   DATA TREE TEST

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	52	52	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 226
OVERALL CORRECT 226 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

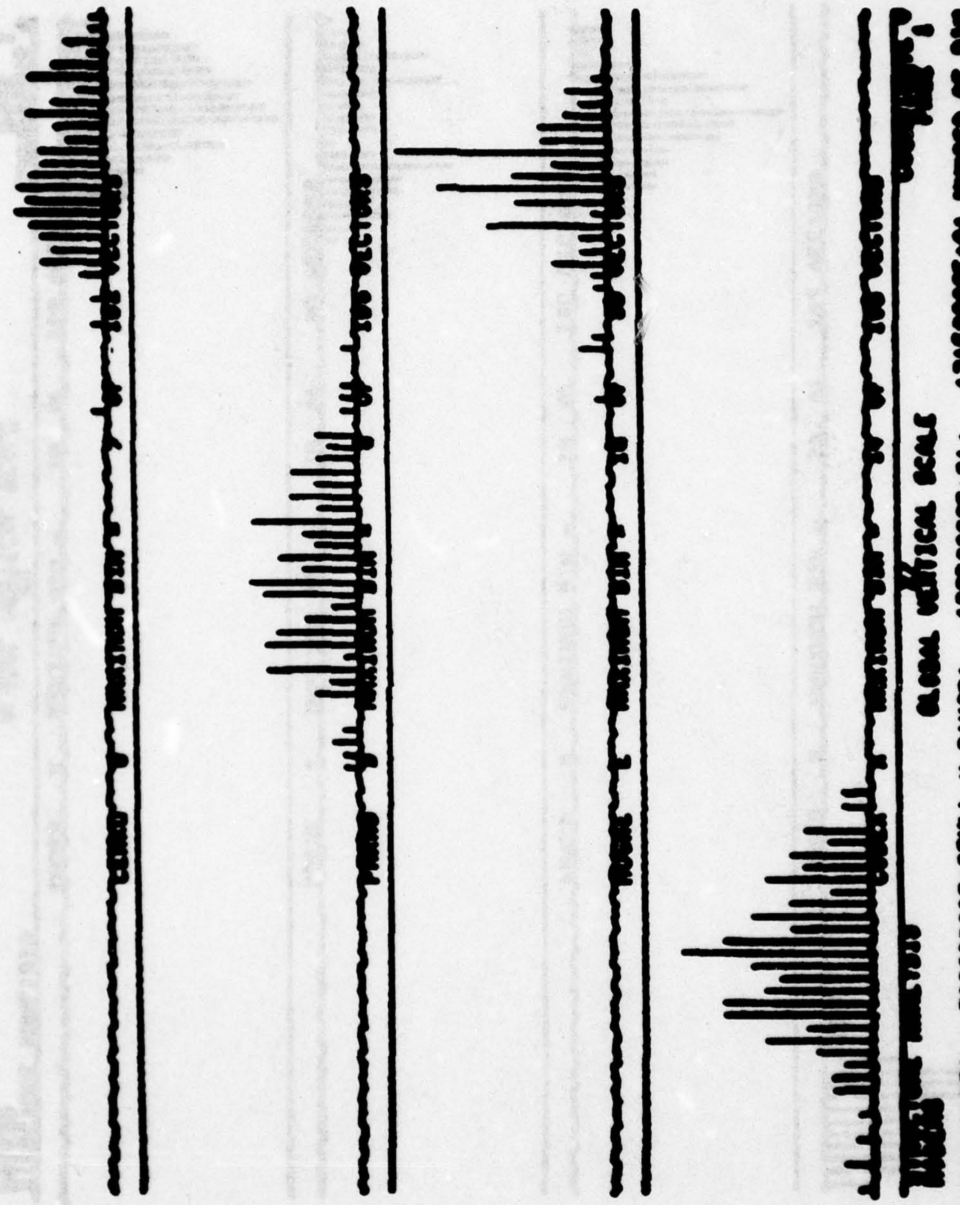
TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 6

FIGURE 24



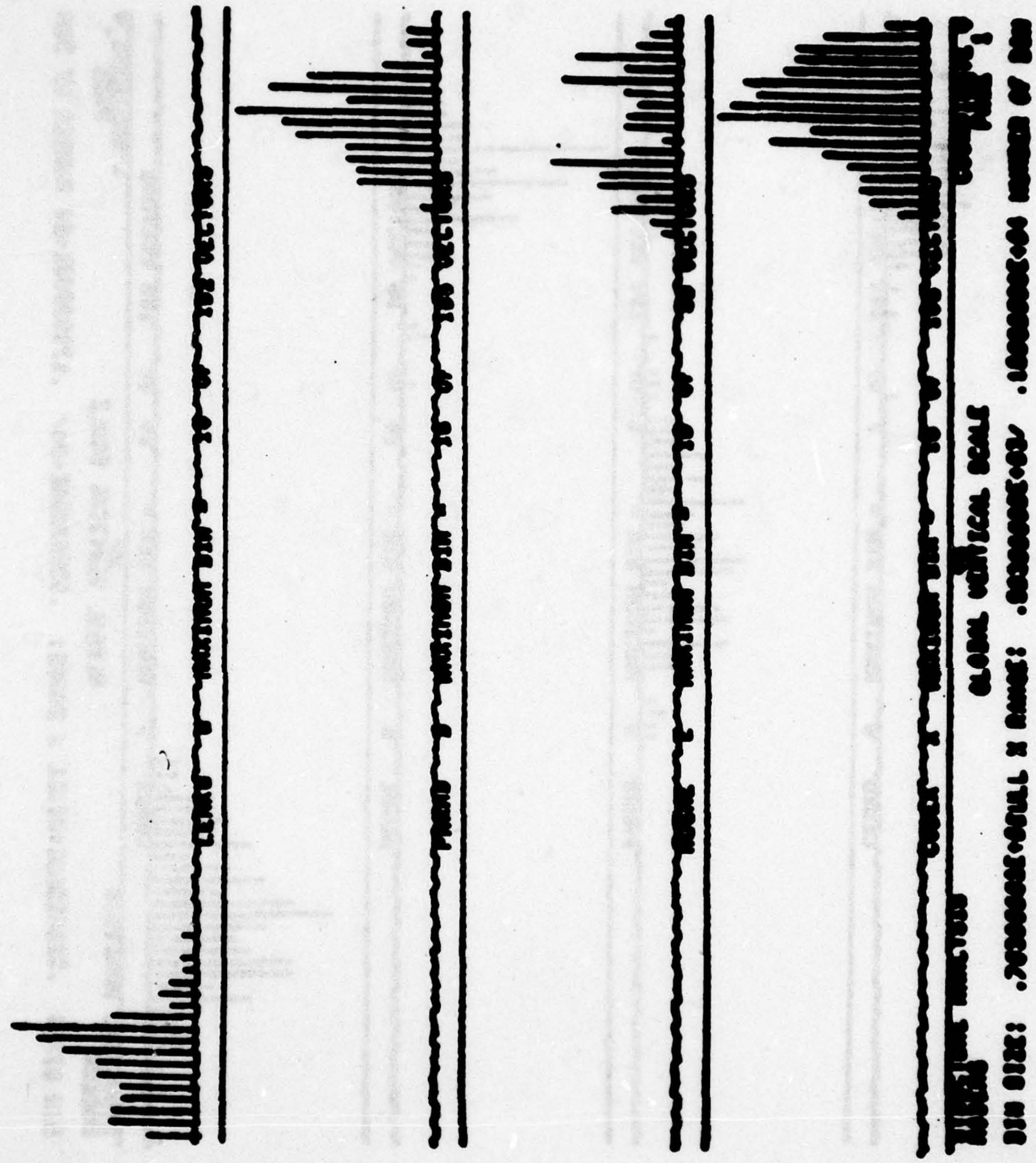
LOGIC TREE STRUCTURE FOR EXPERIMENT 6

FIGURE 25



PROJECTION ON 27th FEATURE DIRECTION

FIGURE 26



PROJECTION ON 46th FEATURE DIRECTION

FIGURE 27

LOGIC TREE YYY DATA TREE DESIGN

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	53	0	0
U	0	0	0	51	0

LOGIC TREE YYY DATA TREE DESIGN

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

```
TOTAL    VECTORS     227
OVERALL  CORRECT     227 FOR 100.00/
OVERALL  ERROR         0 FOR   0.00/
OVERALL  REJECTED      0 FOR   0.00/
```

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 7

FIGURE 28

LOGIC TREE YYY   DATA TREE TEST

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	52	0	0
U	0	0	0	51	0

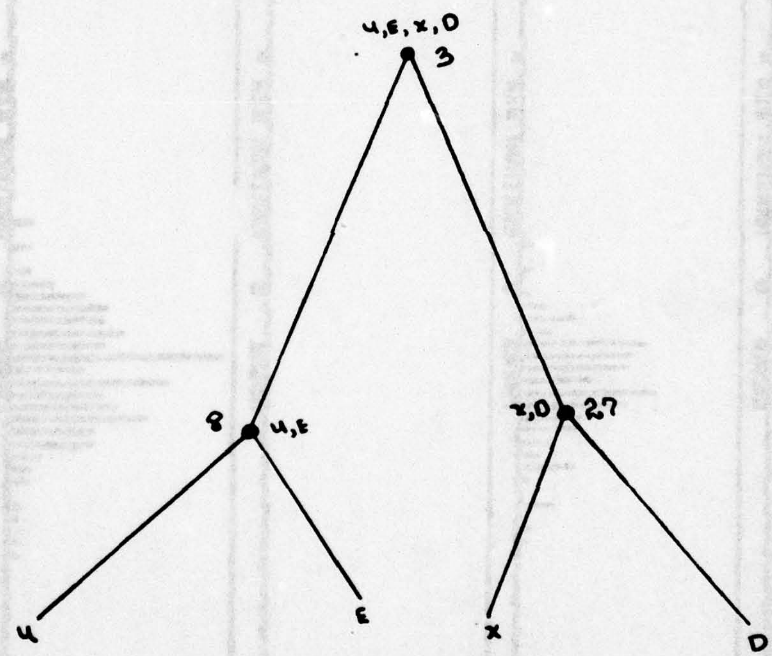
LOGIC TREE YYY   DATA TREE TEST

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	52	52	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

```
TOTAL    VECTORS     226
OVERALL  CORRECT     226 FOR 100.00/
OVERALL  ERROR         0 FOR   0.00/
OVERALL  REJECTED      0 FOR   0.00/
```

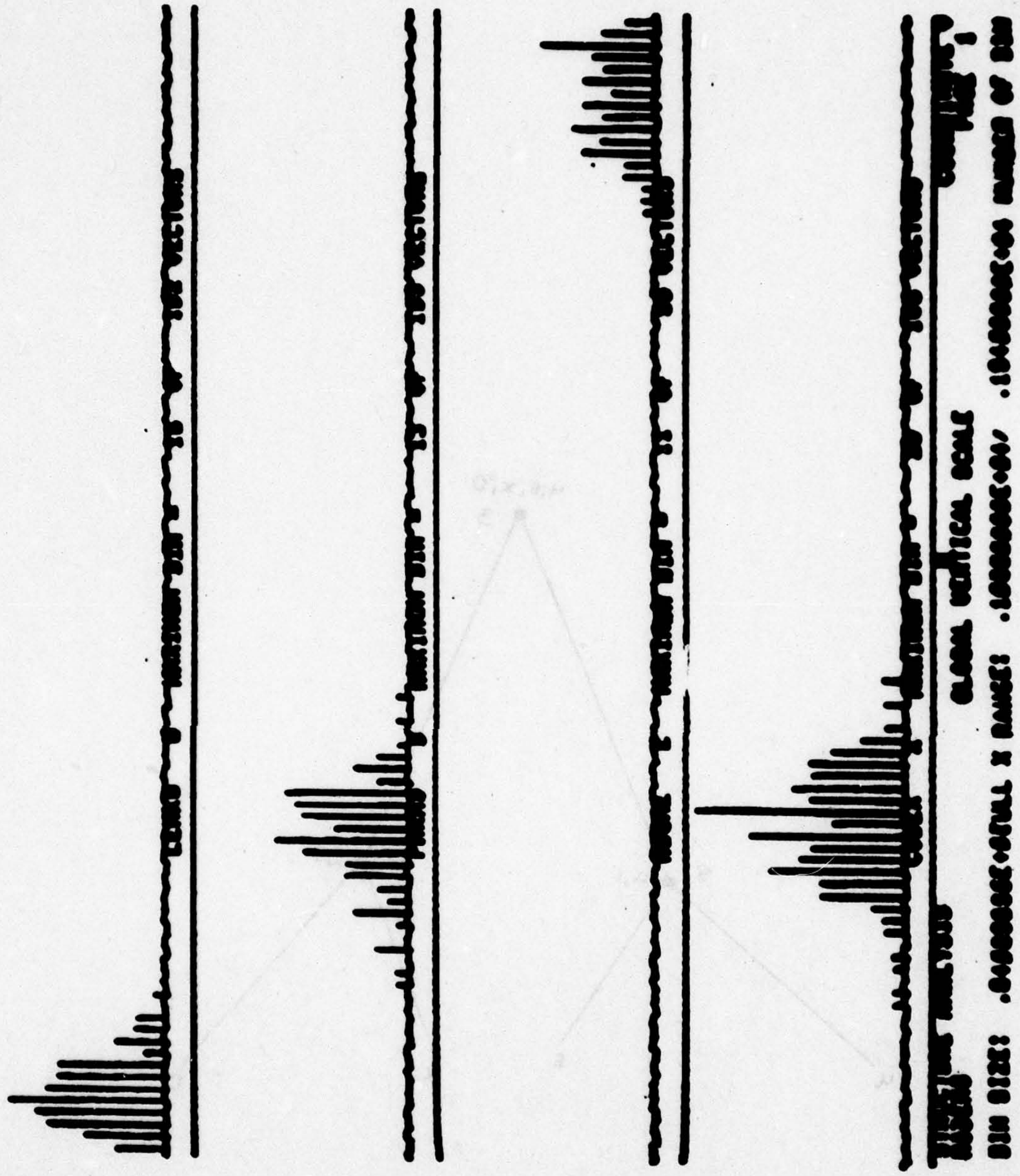
TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 7

FIGURE 29



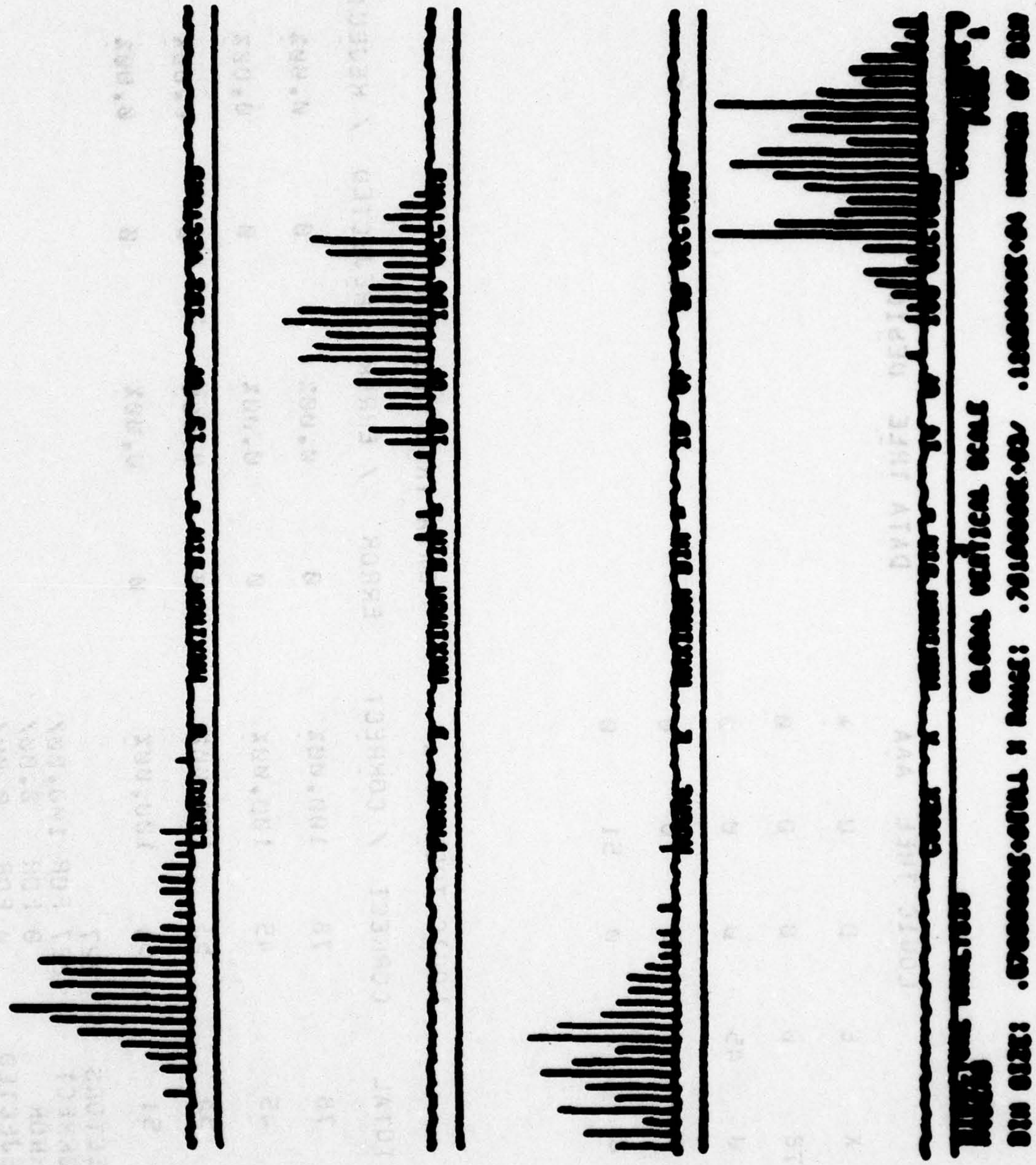
LOGIC TREE STRUCTURE FOR EXPERIMENT 7

FIGURE 30



PROJECTION ON 8th FEATURE DIRECTION

FIGURE 31



PROJECTION ON 3rd FEATURE DIRECTION

FIGURE 32

LOGIC TREE AAA DATA TREE DESIGN

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	53	0	0
U	0	0	0	51	0

LOGIC TREE xxx DATA TREE DESIGN

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

```
TOTAL   VECTORS   227
OVERALL CORRECT   227 FOR 100.00/
OVERALL ERROR       0 FOR   0.00/
OVERALL REJECTED    0 FOR   0.00/
```

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 8

FIGURE 33

LOGIC TREE AAA   DATA TREE TEST

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	52	0	0
U	0	0	0	51	0

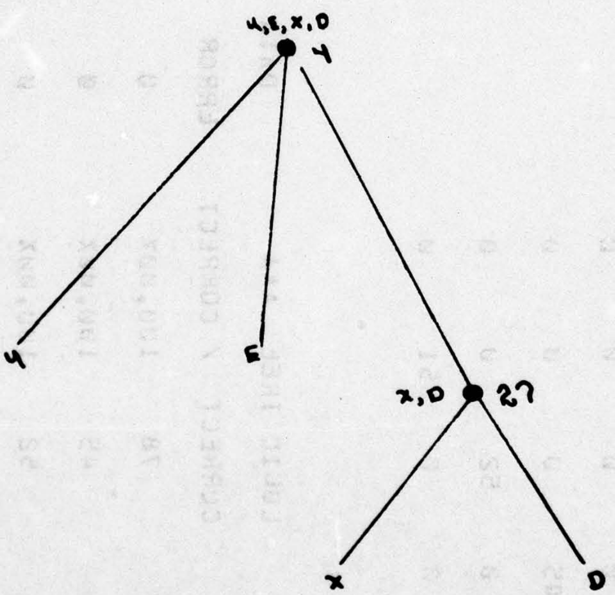
LOGIC TREE AAA   DATA TREE TEST

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	52	52	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

```
TOTAL    VECTORS    226
OVERALL  CORRECT    226 FOR 100.00/
OVERALL  ERROR        0 FOR   0.00/
OVERALL  REJECTED     0 FOR   0.00/
```

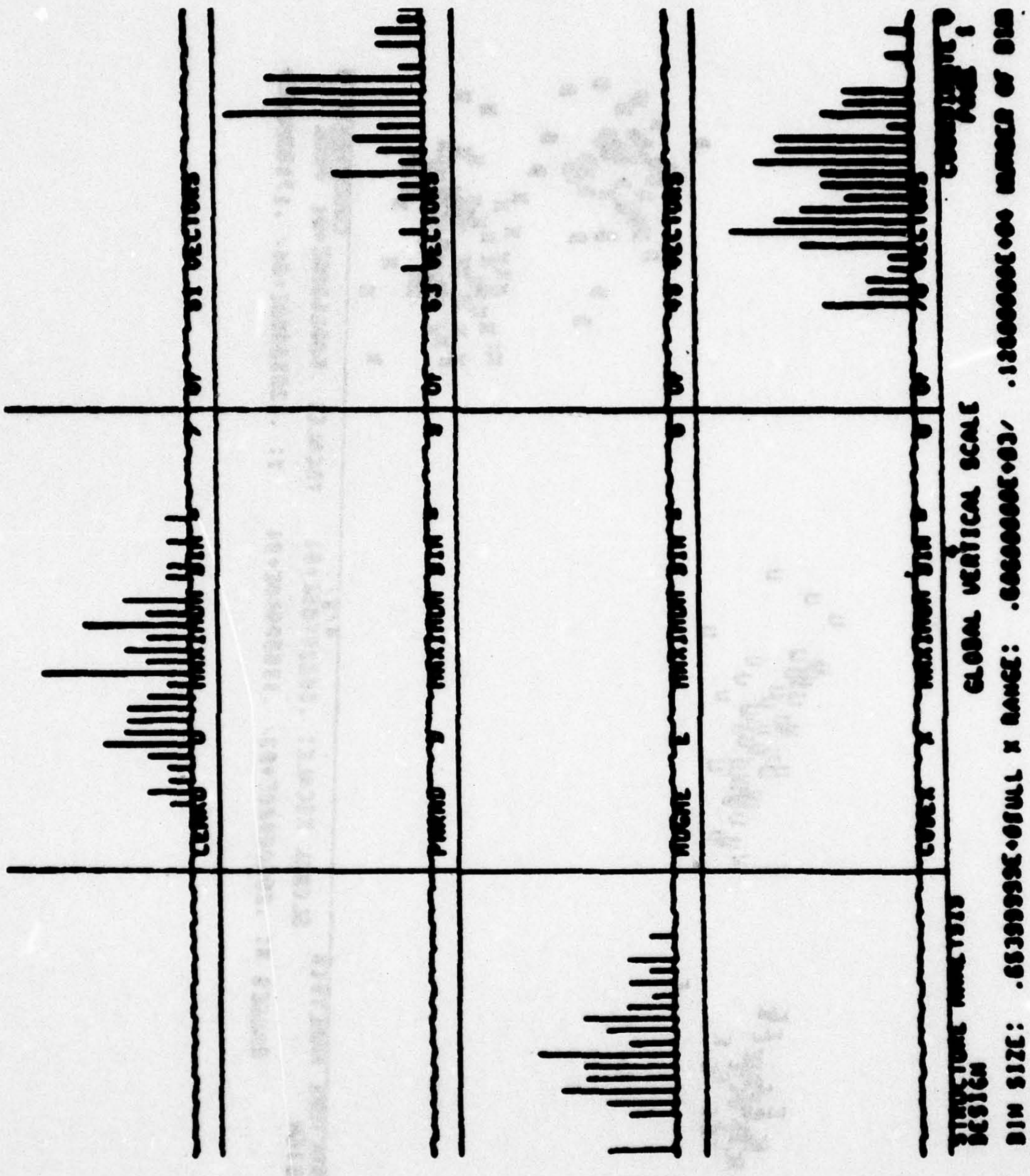
TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 8

FIGURE 34



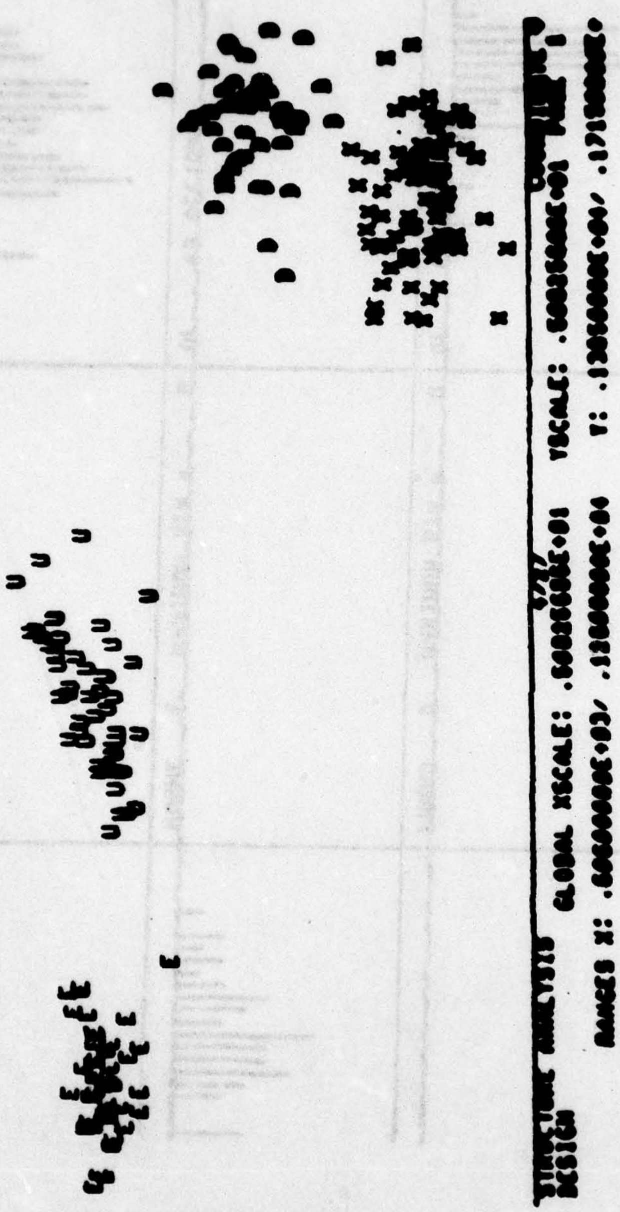
LOGIC TREE STRUCTURE FOR EXPERIMENT 8

FIGURE 35



PROJECTION ON 4th FEATURE DIRECTION

FIGURE 36



TWO-SPACE SCATTER PLOT (MEASUREMENTS 4 AND 27)

FIGURE 37

LOGIC TREE design DATA TREE design

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	53	0	0
U	0	0	0	51	0

LOGIC TREE design DATA TREE design

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	53	53	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 227
OVERALL CORRECT 227 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

DESIGN CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 9

FIGURE 38

LOGIC TREE design DATA TREE TEST

	X	E	D	U	*
X	78	0	0	0	0
E	0	45	0	0	0
D	0	0	52	0	0
U	0	0	0	51	0

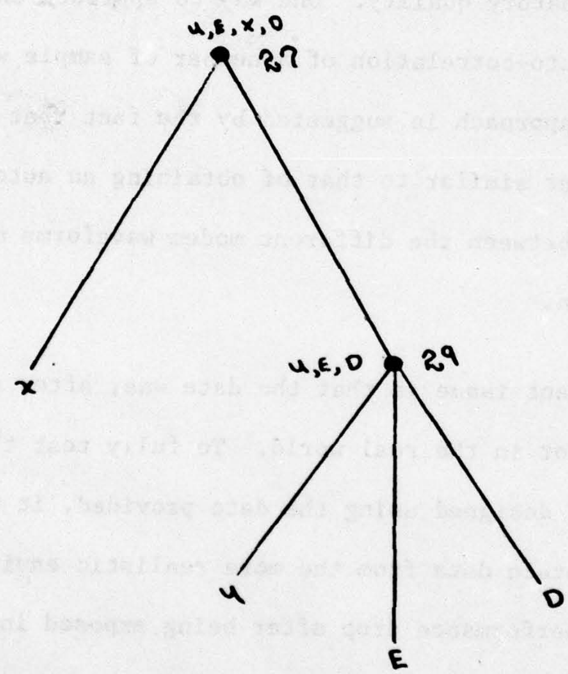
LOGIC TREE design DATA TREE TEST

CLASS	TOTAL	CORRECT	/ CORRECT	ERROR	/ ERROR	REJECTED	/ REJECTED
X	78	78	100.00%	0	0.00%	0	0.00%
E	45	45	100.00%	0	0.00%	0	0.00%
D	52	52	100.00%	0	0.00%	0	0.00%
U	51	51	100.00%	0	0.00%	0	0.00%

TOTAL VECTORS 226
OVERALL CORRECT 226 FOR 100.00/
OVERALL ERROR 0 FOR 0.00/
OVERALL REJECTED 0 FOR 0.00/

TEST CONFUSION MATRIX WITH STATISTICS FOR EXPERIMENT 9

FIGURE 39



LOGIC TREE STRUCTURE FOR EXPERIMENT 9

FIGURE 40

VIII. DISCUSSION AND RECOMMENDATIONS

The success the classifiers had in the problem of modem identification is proportional to the quality of the features provided by the FOBW method. The question remains as to what aspect of the modem's functioning these features were reflecting which provided such excellent discriminatory quality. One way to approach this question is by obtaining the auto-correlation of a number of sample waveforms from each modem. This approach is suggested by the fact that the FOBW method operates in a manner similar to that of obtaining an auto-correlation. Cross-correlation between the different modem waveforms might also provide information.

Another important issue is that the data was, after all, collected in a laboratory, not in the real world. To fully test the performance of the classifiers designed using the data provided, it would certainly be necessary to obtain data from the more realistic environment. Should the classifiers' performance drop after being exposed in the field, information provided by the auto-correlations and cross-correlations obtained previously might suggest other, better features by way of the same FOBW method.

APPENDIX A

PARLAN PROGRAM LISTINGS

LISTING OF PROGRAM BINSEQ

```
10       SUBROUTINE BINSEQ(W1,I10,I20;W2)
20        COMMENT  COMPUTE BINARY SEQUENCE DETERMINED BY A WAVEFORM
30        COMMENT
40        COMMENT  IF A POINT ON W1 IS >0 A (1) RESULTS
50        COMMENT  IF A POINT ON W1 IS <=0 A (0) RESULTS
60        COMMENT  I10,I20 IS THE RANGE TO BE BINARY SEQUENCED
70        COMMENT  IF I20=0 THE WHOLE WAVEFORM WILL BE BINARY SEQUENCED.
80        COMMENT
90        COMMENT  2LT FERNANDEZ  26 APR 78
100       COMMENT
110          IF (I20=0) GOTO 200
120          LET I1=NPT(W1)
130          IF (I20>I1) GOTO 400
140          LET I5=0
150          DO 50 I4=I10,I20
160          LET I5=I5+1
170          LET I9=W1(I4)
180          IF (I9<=0) LET I9=0
190          IF (I9>0) LET I9=1
200          LET W2(I5)=I9
210      50  CONTINUE
220          GOTO 300
230      200 LET I1=NPT(W1)
240          DO 100 I4=1,I1
250          LET I9=W1(I4)
260          IF (I9<=0) LET I9=0
270          IF (I9>0) LET I9=1
280          LET W2(I4)=I9
290      100 CONTINUE
300      300 WAVEND W2
310      400 RETURN
```

LISTING OF PROGRAM FOBW80

```
10    SUBROUTINE FOBW80(W1;V1)
20    COMMENT   PURPOSE:TO COMPUTE THE FREQUENCY OF OCURRENCE OF
30    COMMENT   THE BINARY WORDS "11","10","01", AND "00" WITH DELAYS
40    COMMENT   INPUT---W1:BINARY SEQUENCED WAVEFORM
50    COMMENT   OUTPUT---V1:80 DIMENSIONAL VECTOR (80 FEATURES)
60    COMMENT   2LT FERNANDEZ    11 MAY 78
70          VECTOR V1(80)
80          LET I2=NPT(W1)
90          LET I60=0
100         DO 100 I1=1,200,10
110         LET I3=I2-I1
120         LET I10=0
130         LET I60=I60+1
140         DO 50 I4=1,I3
150         LET I5=W1(I4)
160         LET I6=W1(I4+I1)
170         IF (I5=1 & I6=1) GOTO 25
180         GOTO 50
190   25    LET I10=I10+1
200   50    CONTINUE
210         LET V1(I60)=I10
220   100   CONTINUE
230         DO 200 I14=1,200,10
240         LET I15=I2-I14
250         LET I20=0
260         LET I60=I60+1
270         DO 150 I16=1,I15
280         LET I17=W1(I16)
290         LET I18=W1(I16+I14)
300         IF (I17=1 & I18=0) GOTO 75
310         GOTO 150
320   75    LET I20=I20+1
330   150   CONTINUE
```

```
          LET V1(I60)=I20
      200 CONTINUE
          DO 300 I24=1,200,10
          LET I25=I2-I24
          LET I30=0
          LET I60=I60+1
          DO 250 I26=1,I25
          LET I27=W1(I26)
          LET I28=W1(I26+I24)
          IF (I27=0 & I28=1) GOTO 225
          GOTO 250
      225 LET I30=I30+1
      250 CONTINUE
          LET V1(I60)=I30
      300 CONTINUE
          DO 400 I34=1,200,10
          LET I35=I2-I34
          LET I40=0
          LET I60=I60+1
          DO 350 I36=1,I35
          LET I37=W1(I36)
          LET I38=W1(I36+I34)
          IF (I37=0 & I38=0) GOTO 325
          GOTO 350
      325 LET I40=I40+1
      350 CONTINUE
          LET V1(I60)=I40
      400 CONTINUE
          PRINT V1
          VCEND V1
          RETURN
```

LISTING OF PROGRAM AVERGE

```
      SUBROUTINE AVERGE(W1,I30;W2)
      COMMENT
      COMMENT  PURPOSE : TO COMPUTE THE AVERAGE OF A WAVEFORM AND THEN
      COMMENT  SUBTRACT THIS AVERAGE FROM EVERY POINT OF THE WAVEFORM
      COMMENT  (REMOVE BIAS).  OUTPUT WAVEFORM IS SET TO BE I30
      COMMENT  POINTS LONG.
      COMMENT
      COMMENT
      COMMENT  NPT OF INPUT WAVEFORM MUST BE >=I30.
      COMMENT
      COMMENT  2LT FERNANDEZ     1 JUNE 78
      COMMENT
          LET I1=NPT(W1)
          IF (I1<I30) GOTO 500
          IF (I1>I30) LET I1=I30
          LET I10=0
          DO 100 I2=1,I1
          LET I3=W1(I2)
          LET I10=I10+I3
      100 CONTINUE
          LET I20=I10/I1
          PRINT I20
          DO 200 I4=1,I1
          LET I5=W1(I4)
          LET I6=I5-I20
          LET W2(I4)=I6
      200 CONTINUE
          WAVEND W2
      500 RETURN
```

LISTING OF PROGRAM ANYSEC

```
10      SUBROUTINE ANYSEC(W1,I10,I20,I30,I40;W2)
20      COMMENT   PURPOSE  TO COPY A PART OF A WAVEFORM
30      COMMENT  IF I10 AND I20 ARE BOTH NOT EQUAL TO ZERO THEN THE
40      COMMENT  FIRST I10/I20 FRACTION OF THE WAVEFORM WILL BE COPIED
50      COMMENT
60      COMMENT  IF BOTH I10 AND I20 ARE EQUAL TO ZERO THEN I30 AND I40
70      COMMENT  DENOTE THE RANGE OF THE INPUT WAVEFORM TO BE COPIED.
80      COMMENT
90      COMMENT  2LT FERNANDEZ     14 JULY 78
100     COMMENT
110        IF (I10=0 & I20=0) GOTO 100
120        IF (I20=0) GOTO 500
130        LET I1=NPT(W1)
140        LET I2=I10/I20
150        LET I3=I1*I2
160        DO 10 I4=1,I3
170        LET I5=W1(I4)
180     10 CONTINUE
190        GOTO 400
200     100 LET I1=NPT(W1)
210        IF (I40>I1) GOTO 500
220        LET I8=0
230        DO 300 I4=I30,I40
240        LET I8=I8+1
250        LET I5=W1(I4)
260        LET W2(I8)=I5
270     300 CONTINUE
280     400 WAVEND W2
290     500 RETURN
```

```
10      SUBROUTINE FOBW50(W1;V1)
20      COMMENT   PURPOSE;TO COMPUTE THE FREQUENCY OF OCURRENCE OF
30      COMMENT   THE BINARY WORDS "1,1" , "1,0" WITH DELAYS
40      COMMENT   INPUT---W1:BINARY SEQUENCED WAVEFORM
50      COMMENT   OUTPUT---V1:50 DIMENSIONAL VECTOR (50 FEATURES)
60      COMMENT   2LT FERNANDEZ      3 AUG 78
70          VECTOR V1(50)
80          LET I2=NPT(W1)
90          LET I60=0
100         DO 100 I1=1,25,1
110         LET I3=I2-I1
120         LET I10=0
130         LET I60=I60+1
140         DO 50 I4=1,I3
150         LET I5=W1(I4)
160         LET I6=W1(I4+I1)
170         IF (I5=1 & I6=1) GOTO 25
180         GOTO 50
190     25  LET I10=I10+1
200     50  CONTINUE
210         LET V1(I60)=I10
220     100 CONTINUE
230         DO 200 I14=1,25,1
240         LET I15=I2-I14
250         LET I20=0
260         LET I60=I60+1
270         DO 150 I16=1,I15
280         LET I17=W1(I16)
290         LET I18=W1(I16+I14)
300         IF (I17=1 & I18=0) GOTO 75
310         GOTO 150
320     75  LET I20=I20+1
330     150 CONTINUE
340         LET V1(I60)=I20
350     200 CONTINUE
360         PRINT V1
370         VCEND V1
380         RETURN
```

APPENDIX B

SAMPLE 50 DIMENSIONAL VECTOR
FROM EACH MODEM (CODEX, HUGHES,
PARADYNE, LENKURT, RESPECTIVELY)

.22290000E+04	.21650000E+04	.23040000E+04	.20690000E+04
.14380000E+04	.14240000E+04	.16220000E+04	.12790000E+04
.76700000E+03	.10130000E+04	.11830000E+04	.79800000E+03
.67100000E+03	.11100000E+04	.11220000E+04	.85600000E+03
.11020000E+04	.13900000E+04	.13180000E+04	.11270000E+04
.17180000E+04	.14520000E+04	.14140000E+04	.12920000E+04
.19880000E+04	.13530000E+04	.13930000E+04	.12270000E+04
.18930000E+04	.13160000E+04	.13800000E+04	.11480000E+04
.16820000E+04	.14210000E+04	.14490000E+04	.15770000E+04
.14800000E+04	.15130000E+04	.15180000E+04	.19300000E+04
.14940000E+04	.14890000E+04	.15270000E+04	.19450000E+04
.14770000E+04	.14140000E+04	.15350000E+04	.16020000E+04
.15340000E+04	.14350000E+04	.15840000E+04	.12070000E+04
.15030000E+04	.15140000E+04	.16660000E+04	.10910000E+04
.15290000E+04	.15590000E+04	.16880000E+04	.11720000E+04
.15550000E+04	.15360000E+04	.16080000E+04	.12620000E+04
.15620000E+04	.15080000E+04	.15220000E+04	.11310000E+04
.15610000E+04	.15080000E+04	.15090000E+04	.10720000E+04
.15330000E+04	.15300000E+04	.15390000E+04	.12800000E+04
.15430000E+04	.15190000E+04	.15620000E+04	.16630000E+04
.15280000E+04	.14760000E+04	.15710000E+04	.19550000E+04
.15540000E+04	.14530000E+04	.15750000E+04	.19090000E+04
.15690000E+04	.14430000E+04	.15730000E+04	.15380000E+04
.15600000E+04	.14330000E+04	.15560000E+04	.11980000E+04
.15500000E+04	.14580000E+04	.15440000E+04	.11290000E+04
.85400000E+03	.85000000E+03	.77900000E+03	.86000000E+03
.16450000E+04	.15900000E+04	.14600000E+04	.16500000E+04
.23160000E+04	.20000000E+04	.18980000E+04	.21310000E+04
.24120000E+04	.19020000E+04	.19580000E+04	.20720000E+04
.19010000E+04	.16210000E+04	.17610000E+04	.18000000E+04
.13640000E+04	.15590000E+04	.16650000E+04	.16340000E+04
.10930000E+04	.16580000E+04	.16860000E+04	.16990000E+04
.11870000E+04	.16940000E+04	.16990000E+04	.17780000E+04
.13980000E+04	.15880000E+04	.16300000E+04	.13490000E+04
.16000000E+04	.14950000E+04	.15610000E+04	.99500000E+03
.15860000E+04	.15190000E+04	.15520000E+04	.98000000E+03
.16020000E+04	.15940000E+04	.15440000E+04	.13230000E+04
.15440000E+04	.15730000E+04	.14940000E+04	.17180000E+04
.15740000E+04	.14940000E+04	.14110000E+04	.18340000E+04
.15470000E+04	.14490000E+04	.13880000E+04	.17530000E+04
.15210000E+04	.14720000E+04	.14670000E+04	.16620000E+04
.15140000E+04	.14990000E+04	.15520000E+04	.17920000E+04
.15150000E+04	.14980000E+04	.15640000E+04	.18500000E+04
.15430000E+04	.14750000E+04	.15330000E+04	.16410000E+04
.15330000E+04	.14860000E+04	.15090000E+04	.12580000E+04
.15480000E+04	.15290000E+04	.15000000E+04	.96600000E+03
.15220000E+04	.15520000E+04	.14960000E+04	.10120000E+04
.15070000E+04	.15620000E+04	.14980000E+04	.13830000E+04
.15150000E+04	.15710000E+04	.15150000E+04	.17230000E+04
.15240000E+04	.15450000E+04	.15260000E+04	.17920000E+04